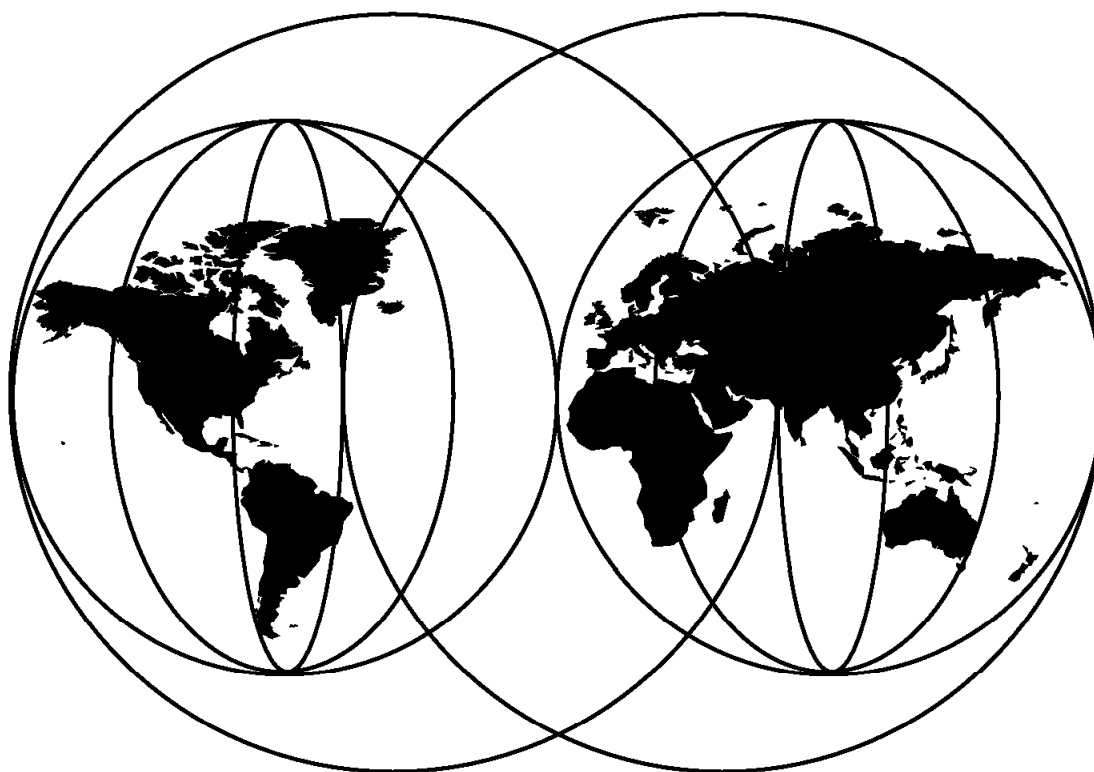




IBM eNetwork Communications Server for OS/2 Warp Version 5.0 Enhancements

Juan R. Rodriguez, Luis Fernando Campos, Faraz T. Khan, Adrian Pardo, Falk Doege



International Technical Support Organization

<http://www.redbooks.ibm.com>

This book was printed at 240 dpi (dots per inch). The final production redbook with the RED cover will be printed at 1200 dpi and will provide superior graphics resolution. Please see "How to Get ITSO Redbooks" at the back of this book for ordering instructions.



International Technical Support Organization

SG24-2147-00

**IBM eNetwork Communications Server
for OS/2 Warp
Version 5.0 Enhancements**

March 1998

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix B, "Special Notices" on page 285.

First Edition (March 1998)

This edition applies to Version 5, Release Number 0 of the IBM eNetwork Communications Server, Program Number 5801-AAR (84H1802) for use with the OS/2 Warp Operating System.

Comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. HZ8 Building 678
P.O. Box 12195
Research Triangle Park, NC 27709-2195

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1998. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Preface	ix
The Team That Wrote This Redbook	ix
Comments Welcome	x

Part 1. Introduction 1

Chapter 1. Introduction	3
1.1 Release Overview	3
1.2 What Communications Server Includes	5
1.3 eNetwork Communications Server for OS/2 Warp Version 5.0 Enhancements	8
1.4 Communications Server Is the Business Solution	9
1.4.1 SNA Gateway Support	10
1.4.2 Host Terminal Emulations	11
1.4.3 Programming Support	11
1.4.4 Communication Features	12
1.5 Communications Manager for Universal Information Access	12
1.5.1 SNA Phone Connect	13
1.5.2 Connectivity Options and Services	13
1.5.3 SNA over TCP/IP Access Nodes and Gateway	14
1.5.4 Sockets over SNA Access Nodes and Gateway	14
1.5.5 TN3270E Server	14
1.5.6 Conformance with SNA	15

Part 2. Client/Server Programming 17

Chapter 2. CPI-C for Java	19
2.1 CPI-C for Java Implementation	19
2.2 CPICJAVA.DLL	20
2.3 What Is Java ?	20
2.4 Benefits	21
2.5 Installation	21
2.5.1 Required Software	21
2.5.2 Installing Feature Install (FI)	21
2.5.3 Installing Java for OS/2 1.1.1	22
2.5.4 Installing CPI-C for Java	25
2.6 Compiling and Executing Sample Java Programs	25
2.7 Compiling and Executing CPI-C for Java Programs	26
2.7.1 Troubleshooting	26
2.7.2 Running JPing	27
2.7.3 Diagnosing CPI-C Errors	29
2.8 Implemented CPI-C Calls on OS/2	29

Part 3. Client Workstations 31

Chapter 3. Windows 95/NT Client Workstations	33
3.1 Overview	33
3.2 Installing the Windows Access Feature from CD	33
3.3 Installing the Access Feature Using Code Server	33

3.4	Installing the Access Feature Using Diskettes	34
3.4.1	Creating Diskettes	34
3.5	Installing Access Feature Using a Response File	35
3.6	SNA Node Configuration	36
3.6.1	Define Node	37
3.6.2	Configure Devices	39
3.6.3	Define Connection	41
3.6.4	Defining a DLUR PU	45
3.6.5	Defining Dependent LUs	46
3.7	Configuring PCOMM Emulation Sessions	48
3.7.1	Using Telnet 3270	48
3.7.2	Using 3270 over LAN (IEEE 802.2)	50
3.8	Monitoring SNA Connections	53
3.9	Troubleshooting	56
3.9.1	Log Viewer	56
3.9.2	Traces	57
3.9.3	GetSense Utility	58
Chapter 4.	OS/2 Client Workstations	59
4.1	OS/2 Access Feature Version 5.0 - Overview	59
4.2	Prerequisite Software	60
4.3	CD-ROM Installation	60
4.4	Installing Using Response Files	60
4.5	Installing Using a Code Server	61
4.6	Installing Using Diskettes	62
4.6.1	Creating Diskettes	62
4.7	OS/2 Access Feature - Sample Installation	63
4.7.1	MPTS Requirements	63
4.7.2	OS/2 Access Feature Installation	63
4.8	OS/2 Access Feature - Sample Configuration	66
4.8.1	Local Node Characteristics	69
4.8.2	Adapter Configuration - Devices	70
4.8.3	Logical Link Configuration - Connections	71
4.8.4	Independent LU Configuration for APPC	76
4.8.5	LU-A Profiles	76
4.8.6	Installing Personal Communications Version 4.2	78
4.9	Monitoring the OS/2 Workstation	80

Part 4. Networking	85
---------------------------	-----------

Chapter 5.	APPN and High-Performance Routing	87
5.1	Overview	87
5.2	LEN and APPN	88
5.3	High-Performance Routing (HPR)	93
5.4	Transmission Groups	93
5.5	Names	94
5.5.1	The Network Accessible Unit	94
5.5.2	Network Identifiers	94
5.5.3	Network Names	94
5.5.4	Network-Qualified Names	94
5.6	Addresses	95
5.6.1	Addresses in Subarea Networks	95
5.6.2	Addresses in APPN Networks	95
5.6.3	Addresses in HPR Networks	96

5.7 Domains	96
5.7.1 APPN Network Node	97
5.7.2 APPN End Node	97
5.7.3 LEN End Node	98
5.7.4 Branch Extender Node (BrNN)	99
5.7.5 Other Node Types	99
5.8 RTP and ANR	103
5.8.1 Rapid Transport Protocol (RTP)	103
5.8.2 Automatic Network Routing (ANR)	105
5.8.3 General APPN/HPR Operation	106
5.8.4 Changes to the APPN Architecture	108
5.9 HPR Base and Options / Enhancements	108
5.9.1 Base Functions for HPR	108
5.9.2 RTP Functions for HPR	110
5.9.3 Control Flows over RTP	110
5.9.4 A Sample HPR Network	111
5.10 Adaptive Rate-Based Flow/Congestion Control	112
5.10.1 ARB Operating Region	112
5.10.2 ARB Principles	113
5.10.3 ARB Algorithm	114
5.10.4 Non-Disruptive Path Switch	115
5.11 Configuration Services	120
5.11.1 HPR Data Link Control	120
5.11.2 Limited Resource	122
5.11.3 HPR Connection Network Support	123
5.11.4 Multi-Link Transmission Groups	123
5.11.5 HPR Link Activation	126
5.11.6 Link Data Traffic	128
5.11.7 After Link Activation	129
5.11.8 Link Failure Detection	129
5.12 HPR and BrNNs	129
5.12.1 RTP Connections Terminating on a BrNN	130
5.13 HPR Configuration	130
5.13.1 LAN and Frame Relay - HPR Configuration	131
5.13.2 SDLC	134
5.13.3 NDF Configuration	136
5.14 Monitor HPR Connections	138
5.14.1 HPR Commands	138
Chapter 6. HPR over WAN - SDLC Support	141
6.1 Installation of SDLC Protocol	141
6.1.1 WAC Adapter - Installation	142
6.1.2 Configure SDLC Protocol	147
6.2 Troubleshooting	154
Chapter 7. Branch Extender Node	157
7.1 Overview	157
7.1.1 Necessity	159
7.1.2 Requirements	160
7.2 BrNN - Design Overview	161
7.3 Restrictions	163
7.3.1 Connection Networks and BrNN	164
7.3.2 HPR in a BrNN	165
7.3.3 DLUR and BrNN	165
7.3.4 Cryptography and BrNN	165

7.4 Connecting BrNNs	166
7.5 BrNN Configuration	166
7.5.1 Configuration Options	166
7.5.2 Response File - BrNN Configuration	169
7.5.3 NDF File - BrNN Configuration	170

Part 5. AnyNet Support 171

Chapter 8. AnyNet SNA over TCP/IP - TCP62 API	173
8.1 Overview	173
8.1.1 Writing TCP62 Programs	174
8.1.2 TN62API() Entry Point	174
8.1.3 Migration Considerations	174
8.1.4 Dynamic Name Generation	175
8.2 TCP62 API Support	177
8.2.1 START_TN62	177
8.2.2 Supplied Parameters	177
8.2.3 Returned Parameters	180
8.2.4 STOP_TN62	180
8.2.5 Supplied Parameters	181
8.2.6 Returned Parameters	181
8.2.7 DEFINE_LOCAL_LU_TN62	181
8.2.8 Supplied Parameters	182
8.2.9 Returned Parameters	184
8.2.10 DEFINE_PARTNER_LU_TN62	184
8.2.11 Supplied Parameters	185
8.2.12 Returned Parameters	185
8.2.13 TCP62 Return Codes	185
8.3 TCP62 Configuration	187
8.3.1 HOSTS Files	188
8.3.2 SNA Local Node Configuration in Server B and OS/2 Workstation	189
8.3.3 Executing TN62TEST.EXE from the OS/2 Command Line	192
8.3.4 TCP62 Status Monitoring	192
8.3.5 APING from Server B to OS/2 Workstation	193
8.3.6 TCP62 Status Monitoring after Executing the APING Command	193
8.3.7 TN62TEST.EXE Command Syntax	193
8.3.8 RSP File Produced by TCP62	195

Part 6. Dependent LU Support 197

Chapter 9. SNA Gateway in Branch Extender Nodes	199
9.1 SNA Gateway Support	199
9.2 SNA Gateway Configuration	200
9.2.1 SNA Gateway - Hosts and Host LU Pools	202
9.2.2 SNA Gateway - Implicit Workstations Using the Gateway	203
9.2.3 SNA Gateway - Explicit Workstations Using the Gateway	203
9.2.4 BrNN Local Node Configuration	204
9.2.5 DLUR PU - Configuration	205
9.2.6 Host Links	207
9.2.7 Host LU Pools	207
9.2.8 Downstream Workstations	209
 Chapter 10. TN3270E Server in Branch Extender Nodes	 213

10.1 TN3270E Scenario	213
10.2 TN3270E Server Configuration	214
10.2.1 SNA Local Node Characteristics	215
10.2.2 SNA Dependent LU Server Definitions	216
10.2.3 TN3270E Server Parameters	217
10.3 Workstation Configuration	220
10.4 TN3270E Server Administration	220
Chapter 11. Host On-Demand	225
11.1 Host On-Demand - Overview	225
11.1.1 Required Software - Server	227
11.1.2 Required Software - Clients	227
11.2 Host On-Demand Installation	228
11.2.1 ICSS Request Routing for HOD	229
11.2.2 IBM Communications Server for OS/2 Configuration	231
11.2.3 Host On-Demand Configuration and Additional Information	231
11.2.4 Host On-Demand Session	233

Part 7. Operations and Administration 237

Chapter 12. Web-Based Administration	239
12.1 Required Software	239
12.2 Required Hardware	240
12.3 Security	240
12.4 Installation	240
12.4.1 Installing the IBM ICSS 4.2.1	240
12.4.2 Installing Web Administration	244
12.5 Customizing the FFST/2	247
12.6 How to Use Web-Based Administration	247
12.6.1 Web Administration Pages	248

Part 8. Problem Determination 255

Chapter 13. Problem Determination	257
13.1 Troubleshooting Information	257
13.2 FFST/2	258
13.3 APPNT and APPNF	259
13.4 Trace Services	261
13.5 Ring 0 - Traces	263
13.5.1 Setting Ring 0 Traces	263
13.5.2 Formatted ANR Traces	264
13.5.3 Trace Spooling	265

Appendix A. TCP62 API Sample Program	267
A.1 TCP62 Sample Program - Header File	267
A.2 TCP62 Sample Program	270

Appendix B. Special Notices	285
------------------------------------	-----

Appendix C. Related Publications	287
C.1 International Technical Support Organization Publications	287
C.2 Redbooks on CD-ROMs	287
C.3 Other Publications	287

How to Get ITSO Redbooks	289
How IBM Employees Can Get ITSO Redbooks	289
How Customers Can Get ITSO Redbooks	290
IBM Redbook Order Form	291
 Index	 293
 ITSO Redbook Evaluation	 295

Preface

This redbook helps you to understand the latest release of IBM eNetwork Communications Server for OS/2 Warp Version 5.0. It focuses on the new networking support implemented in this product, and helps you plan, install, and configure the new functions quickly in a wide variety of environments.

In this redbook you will find information on the installation of SNA nodes such as End Nodes, Network Nodes and the new Branch Extender Node.

You will also find numerous configuration examples showing ways to set up the SNA gateway and the TN3270E server when connected to subarea or APPN networks using WAN protocols.

Some knowledge of previous releases of the IBM eNetwork Communications Servers, as well as a basic knowledge of networking concepts and terminology used in the Systems Network Architecture (SNA) and TCP/IP is assumed.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Raleigh Center.

Juan R. Rodriguez is a Networking Specialist at the IBM ITSO Center, Raleigh. He received his M.S. degree in Computer Science from Iowa State University, writes extensively and teaches IBM classes worldwide on areas such as networking and data security. Before joining the ITSO two years ago, he worked at the IBM laboratory in Research Triangle Park (North Carolina, USA) as a designer and developer in networking products.

Luis Fernando Campos is a networking specialist for the Product Support Services Division in IBM Brazil. He has three years of experience in networking products such as TCP/IP, PCOMM, CM/2, VTAM, NetWare for SAA, routers/gateways and LAN communication protocols. His areas of expertise include LAN servers such as Novell NetWare and OS/2 LAN Server.

Faraz T. Khan is a Pre-Sales IT-Specialist for the IBM Networking Division in IBM Australia. He has experience in Campus Networking Hardware and Software Products. His areas of expertise include TCP/IP architectures, SNA/APPN/HPR and LAN communication protocols. He holds a Masters by Research in Computer Engineering from the University of New South Wales in Sydney, Australia. He has written and submitted papers to the IEEE Transactions on Neural Networks in the field of globally convergent feedforward neural networks.

Adrian Pardo is a network specialist in IBM Argentina. He has 21 years with IBM and has had several staff and field engineering positions. His areas of expertise include OS/2, Communications Server, LAN Server and recently Windows NT.

Falk Doege is a Support Engineer in IBM Germany. Since 1995 he has been working in the IBM networking software support in the Computer Service Gesellschaft (CSG), an IBM Germany service organization. His areas of expertise include PCOMM, Communications Server, LAN Server, LAN Distance, OS/2, Windows NT and Novell NetWare. He is a Microsoft Certified Professional

for Windows NT 4.0 and an IBM Certified Professional for eNetwork. He holds a degree in Electro Engineering from Ingenieur Hochschule Dresden, Germany.

Thanks to the following people for their invaluable contributions to this project:

John Mitchell
IBM Research Triangle Park, North Carolina, USA

Mike Kawalec
IBM Research Triangle Park, North Carolina, USA

Suvas Shah
IBM Research Triangle Park, North Carolina, USA

Harold Fanning
IBM Research Triangle Park, North Carolina, USA

Chuck Fricano
IBM Research Triangle Park, North Carolina, USA

Jill Bodine
IBM Research Triangle Park, North Carolina, USA

Shawn Walsh
IBM ITSO Raleigh Center, North Carolina, USA

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 295 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:

For Internet users	http://www.redbooks.ibm.com/
For IBM Intranet users	http://w3.itso.ibm.com/

- Send us a note at the following address:
redbook@us.ibm.com

Part 1. Introduction

Chapter 1. Introduction

Today's personal computers, combined with the power of OS/2, offer new ways of networking directly from your desktop. For example, you can communicate and share resources with someone in the next room, the next city, or on another continent. At the same time, an application running on your workstation can be accessing data from a host computer while distributing data to another workstation on a local area network (LAN). You can even have a conference from your personal computer with live video and interactive data exchanges across a network. Distributed processing and client/server applications can also share the personal computer with terminal emulator programs that connect you to applications running on a host computer. These are only a few of the applications of personal networking from your desktop workstation.

1.1 Release Overview

Communications Server brings the power of personal networking to your workstation. Whether it's for host terminal emulation, client/server and distributed applications, or connectivity across local and wide area networks (LANs and WANs), Communications Server offers a robust set of communications, networking, and system management features.

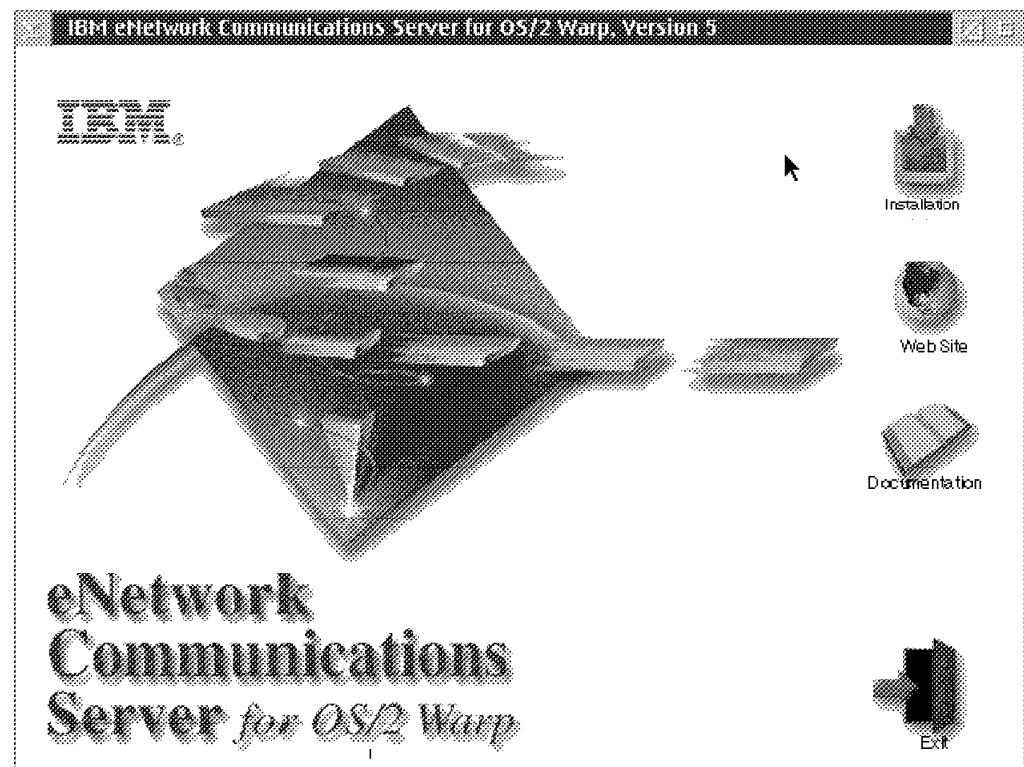


Figure 1. eNetwork Communications Server for OS/2 Warp Version 5.0

For true networking flexibility, a wide range of connectivity services and options are provided. With Communications Server, workstations and gateways can communicate using several communication protocols such as SDLC, IDLC, X.25, TCP/IP, NetBIOS, and IEEE 802.2. These protocols can be used over coaxial, twinaxial, LAN (Token-Ring, Ethernet, PC Network, FDDI, Frame Relay, and ATM),

switched lines and nonswitched lines, and ISDN connections. Mobile users can directly access their host system or another Communications Server via public switched telephone networks and via wireless and cellular networks.

The versatility of Communications Server extends to the types of applications that can be supported. Communications Server supports a wide variety of application programming interfaces (APIs) and protocols that are ideal for client/server applications and distributed processing. The Common Programming Interface for Communications (CPI Communications) and advanced program-to-program communication (APPC) support makes Communications Server the ideal communications platform for peer application environments.

Advanced Peer-to-Peer Networking (APPN) adds functions, formats, and protocols that enhance the managing of an SNA network and the usability of APPC applications running in the network. The APPN network nodes provide directory, route selection, and management services to end nodes. The end nodes provide these services to users at their workstations.

Communications Server lets workstation users and applications communicate with other workstations and host computer applications. In addition, Communications Server provides versatility and flexibility to workstations communicating across SNA and TCP/IP networks. Communications Server does this by providing a comprehensive set of services and functions, all in one package.

Communications Server contains a powerful gateway facility. The gateway allows workstations on a local area network to connect to host computers through one set of adapters and physical connections. The Communications Server gateway supports various OS/2, DOS, and Windows workstations. Remote workstations can also dial in to this gateway and then use a common high-speed link to the host. The gateway types provided are:

- **LAN gateways** enable Novell NetWare Internet Packet Exchange (IPX) applications and NetBIOS applications running on a LAN to communicate over SNA and TCP/IP networks.
- **SNA gateways** enable OS/2 workstations to act as communications controllers between multiple workstations and one or more SNA hosts.
- **SNA over TCP/IP gateways** connect SNA and TCP/IP networks to enable communications across these networks.
- **Sockets over SNA gateways** enable workstations that have TCP/IP configured to communicate with workstations that have the Sockets over SNA access feature enabled.
- **Paired Sockets over SNA gateways** enable workstations that have TCP/IP configured to communicate over an SNA network.
- **TN3270E Server gateways** enable TN3270 and TN3270E-compliant clients to gain access to host applications.
- **Host On-Demand** provides fast and easy access to host information from intranets and the Internet.

Communications Server implements the multiprotocol transport networking (MPTN) architecture in its SNA over TCP/IP and Sockets over SNA functions. Application programs can communicate independently of the underlying network transport protocol. Thus, you can combine networks without losing application support.

The SNA over TCP/IP access node function allows SNA applications to communicate with each other over an IP network. The SNA over TCP/IP gateway function lets SNA applications on existing SNA networks communicate, without change, with SNA applications on the IP network.

The Sockets over SNA access node function allows socket applications to communicate with each other over an SNA network. The Sockets over SNA gateway function allows socket applications on existing TCP/IP networks to communicate, without change, with socket applications on an SNA network. Two Sockets over SNA gateways can be used to connect isolated TCP/IP networks to an SNA network, thereby allowing socket applications running on the separate TCP/IP networks to communicate through the SNA network.

Communications Server includes tools to simplify the administration of SNA networks. These include system and network alerts, message and error logs, trace and dump files, and programming capabilities that allow you to configure and manage networks and build sophisticated management programs. Complementary network and system management products can access all of these facilities.

Communications Server protects your investment in Communications Manager applications by providing upward compatibility for applications written to the CM APIs of OS/2 Extended Edition, Extended Services for OS/2, Communications Manager/2, Communications Server, and AnyNet. You also have the ability to use the SNA and sockets applications that you used with AnyNet for OS/2 products with Communications Server.

For customers that used the Communications Manager/2 emulator, IBM provides terminal emulator products, such as the Personal Communications family of emulators, available as complementary products on OS/2 and Windows. With these emulator products, desktop workstations can emulate several types of host-attached terminals, including:

- 3270 terminal emulation for System/370 and System/390 hosts
- 5250 terminal emulation for Application System/400 (AS/400) computers

1.2 What Communications Server Includes

The Communications Server package contains the following products:

1. eNetwork Communications Server for OS/2 Warp Version 5.0

The Communications Server provides:

- APPN network and end node
- Branch Extender
- Frame relay
- High-Performance Routing (HPR)
 - Automatic Network Routing (ANR)
 - Rapid Transport Protocol (RTP)
 - Multi-link Transmission Groups (MLTG)
- Installation and configuration support
- Local and wide area connectivity support

- Multiprotocol gateway support
 - LAN
 - SNA
 - SNA over TCP/IP
 - Sockets over SNA
 - TN3270E Server
- Multiprotocol Transport Services (MPTS)
- IBM Personal Communications AS/400 and 3270 - APPC/LUA Entry Level (Personal Communications Lite)
- SNA network support and systems management
- WIN-OS/2 CPI-C support
- 32-bit OS/2 application programming interfaces (APIs)

2. OS/2 Access Feature, Version 5

OS/2 Access Feature provides 32-bit OS/2 API support. OS/2 Access Feature enables applications that are written to the Sockets, APPC, CPI-C, and LUA APIs to run unchanged over either SNA or TCP/IP local and wide area networks. OS/2 Access Feature also provides applications with local and wide area connectivity support. This is the recommended version for workstations running OS/2 Warp 3.0 or later.

3. The OS/2 Access Feature provides:

- APPN end node
- Frame relay
- HPR
 - RTP
 - MLTG
- Installation and configuration support
- Local and wide area connectivity support
- Multiprotocol access node support
 - SNA over TCP/IP
 - Sockets over SNA
- MPTS
- SNA network support and systems management
- WIN-OS/2 CPI-C support
- 32-bit OS/2 application programming interfaces (APIs)

4. OS/2 Access Feature, Version 4.0

The OS/2 Access Feature, Version 4.0 provides 32-bit OS/2 API support similar to that in OS/2 Access Feature, Version 4.1. You must use this version for workstations running OS/2 2.11.

5. Windows Access Feature Version 4.1

The Windows Access Feature is a bundle of two products that provide APPC programming support and enable APPC applications to run unchanged over either SNA or TCP/IP local and wide area networks. The products are:

- APPC Networking Services for Windows Version 1.0.2
- AnyNet APPC over TCP/IP for Windows Version 1.0

The Windows Access Feature also includes the LAN Support Program.

6. Host On-Demand

IBM Host On-Demand is a Java-based solution that incorporates industry-standard Telnet 3270 (TN3270) protocols. It provides a high-performance, low-cost solution for intranet and Web users who need occasional access to their central computer applications or databases from any Java-enabled end-user platform. Host On-Demand includes a number of powerful features, such as:

- Emulator functions on demand
- Java-based implementation
- Customized 3270 windows
- Multiple sessions
- Persistent connections
- Platform flexibility
- Security
- Investment protection

7. Web Administration Console

The Web Administration Console allows remote administration of Communications Server using a Web browser. When installed with an Internet Connections Server, this tool allows administrators to view subsystem statistics, start and stop the server, manage links, and provides a command line interface for accessing other components like AnyNet LAN Gateway, TCP/IP, and OS/2 resources.

8. Windows 95/NT Access Feature

The Windows 95 and Windows NT Access Feature brings the power of personal networking to workstations by exploiting the networking capabilities of Windows NT and Windows 95 to provide a variety of connectivity options supporting local area network (LAN) and wide area network (WAN) environments.

The CD-ROM also contains:

- Product files
- Product install guide
- Documentation
- Productivity aids
- Sample response files
- API support files
- Adapter device drivers

Throughout this book, the name eNetwork Communications Server for OS/2 Warp Version 5.0 refers to both Communications Server and OS/2 Access Feature.

1.3 eNetwork Communications Server for OS/2 Warp Version 5.0 Enhancements

eNetwork Communications Server for OS/2 Warp Version 5.0 is based on the very popular Communications Server Version 4.1. If you are familiar with Communications Server Version 4.1, you might want to review some of the new functions and enhancements described below.

Refer to Communications Server Version 4.1 Enhancements (SG24-4916) for enhancements made in the previous release.

Refer to Communications Server Version 4.0 Enhancements (SG24-4587) for updates made in Communications Server 4.0 as enhancements to Communications Manager/2.

- High Performance Routing over wide area networks

HPR provides improved performance, reliability, and availability. This release extends the types of links over which HPR connections can be established. It includes support for WAN data link controls such as SDLC and other data link controls supported by GDLC with a vendor adapter. In addition, it includes support for the case where a logical link appears to be a LAN link, but is actually transparently bridged or routed over a WAN (such as for Frame relay). Note that LAN DLC for PCNET does not support HPR.

Enhancements for this release include:

- Support for Data Link Controls other than IEEE 802.2 over LANs

This release includes support for SDLC, GDLC and IDLC.

- Support for link-level error recovery protocols (ERP) on a LAN

Error recovery protocol (ERP) is a method of detecting a lost packet at the destination and recovering by asking the source to retransmit the packet.

This release enables you to select ERP on each link, if you choose. Since Rapid Transport Protocol (RTP) detects and recovers lost packets at a connection's end points, you can use either ERP links or non-ERP links when you build your network. This enables you to specify link-level ERP on links that have a high rate of packet loss, and to maximize throughput on other links by specifying that they not use link-level ERP.

- Support for multi-link transmission groups

This release provides enhancements to include HPR support for APPN multi-link transmission groups (MLTG). MLTG support is useful when additional capacity is needed during peak periods. Links can be added and removed from an MLTG dynamically without affecting any network applications. This can lower costs by allowing you to add the extra capacity only when needed. In addition, MLTG support can lower costs by allowing several circuits to be used together to achieve the capacity needed, a technique that lessens the impact of carrier pricing policies where very high speed links are significantly more expensive than the equivalent combination of lower speed links.

- Branch Extender

Without the Branch Extender, problems occur if an APPN topology subnetwork exceeds the smallest network node's topology database (TDB) storage. This could limit network size to as few as 150 network nodes. (TDB size is a function of the number of network nodes and links, as well as the frequency of state changes.) Many administrators would prefer to install 1500 or more small network nodes as gateways to branch offices, with a LAN at each branch office that is configured as a connection network.

The Branch Extender enables you to interconnect a branch office with LANs, end nodes and low end network nodes with dependent and independent LUs, and PUs such as teller machines, to one or several WANs.

- Host On-Demand

IBM Host On-Demand is an Internet-to-SNA interconnectivity solution that provides 3270 application discovery and access through the World Wide Web (WWW). Web users needing access to host applications such as public catalogs, software applications, databases, or other resources can use Host On-Demand from inside a Java-enabled Web browser to access central computer data.

- Windows 95 and Windows NT Access Feature

This function provides communications to host machines and other connections from Windows 95 or Windows NT client workstations.

This feature includes support for Advanced-Peer-to-Peer Networks (APPN) as an end node, and uses the advanced network features such as High Performance Routing (HPR) and Dependent LU Requester (DLUR).

This feature also includes AnyNet SNA over TCP/IP, which allows client/server SNA applications to communicate over a TCP/IP network.

- Web Administration Console

This feature allows remote administration of Communications Server using a Web browser. Web Administration Console allows administrator's to view subsystem statistics, start and stop the server, manage links, and provides a command line interface for accessing other components, such as AnyNet LAN Gateway, TCP/IP, and OS/2 resources.

1.4 Communications Server Is the Business Solution

Communications Server is the right server platform for your business, as it allows you to tap into the full set of IBM solutions.

The powerful SNA gateway facility allows many workstations to connect to a host computer through a single Communications Server workstation, making it possible to bring host resources to many users while keeping adapter and connection costs down. From the workstations, the IBM Personal Communications family of emulators provide access to host applications. Communications Server also contains a rich set of application programming interfaces (APIs) and support for a wide range of connected environments.

Communications Server enables you to:

- Make application decisions based on business needs, independent from existing network protocols and platforms.

- Access the information you need, when you need it, from a large computer or LAN whether you're at home, on the road, or in a customer's office.
- Improve your network systems management through consolidated traffic and reduced need for parallel networks.
- Prioritize different kinds of data traffic for TCP/IP and SNA applications.

The following sections describe:

- SNA gateway support
- Host terminal emulation
- Programming support
- Communication features

1.4.1 SNA Gateway Support

Communications Server provides a full-function Systems Network Architecture (SNA) gateway. The gateway allows multiple LAN-attached workstations to access System/370 or System/390 hosts through one or more physical connections to one or more hosts. This helps reduce the cost per workstation of host connections.

The Communications Server gateway supports the SNA protocols LU 0, 1, 2, 3, and dependent LU 6.2 (APPC). With the SNA over TCP/IP function, downstream workstations can now communicate with the SNA gateway over an IP network. The gateway also supports LU 0, 1, 2, or 3 to an AS/400 host using SNA pass-through. The AS/400 host passes the data through to a System/390 host.

A gateway also acts as a protocol converter between workstations attached to a LAN and a host WAN link.

An SNA gateway enables supported workstation applications to access remote applications on a subarea network without requiring a separate direct connection to each host from each workstation.

The LUs defined in the gateway can be dedicated to a particular workstation or pooled among multiple workstations. Pooling allows workstations to share common LUs, which increases the efficiency of the LUs and reduces the configuration and startup requirements at the host. You can also define multiple LU pools, where each pool is associated with a specific application. And you can define common pools that are associated with multiple hosts. When a link is defined through the gateway between a workstation and the host, the LU is activated when the session is established and returned to the pool for access by other workstations when the session is ended.

You can also configure an SNA gateway to automatically stop a session after a specified period of inactivity if there are other workstations waiting. This procedure helps increase the number of shared logical units that are available.

In addition, an SNA gateway can support the transmission of network management vector transports (NMVTs) between the gateway and the host. For example, commands coming from the NetView program in the host are received in the gateway and can be passed to and used by another application on the gateway, such as IBM LAN Network Manager.

Each host views the SNA gateway as an SNA PU 2.0 node, supporting one or more LUs per workstation. As far as the host is concerned, all LUs belong to the SNA gateway PU. The SNA gateway can have multiple host connections simultaneously and can direct different workstation sessions to specified hosts. However, only one host (and it must be on a link with a control point PU) can act as the focal point, and the control point name is appended to all NMVTs routed through the gateway.

To the supported workstations, the SNA gateway looks like an SNA PU 4 communications controller and forwards such host requests as BIND and UNBIND. The workstation LUs are not aware of the SNA gateway. The SNA gateway, however, is aware of all LUs at the workstations.

The SNA gateway is a special type of PU 2.0. As long as a dependent workstation is inactive, the SNA gateway implements the LU functions for the workstation, just as an ordinary PU 2.0 would. However, as soon as a workstation is online to the host, the SNA gateway allows the workstation to implement LU functions and merely passes data between workstations and the host.

1.4.2 Host Terminal Emulations

Connections from the workstation to the host, called emulator sessions, allow the workstation to perform host computer functions, such as data processing, file access and host printing, in the same manner as 3270 and 5250 display terminals.

The terminal emulator previously provided in Communications Manager/2 has been removed. Terminal emulation for administrative use at the server is provided by Personal Communications Lite. Workstation terminal emulation is provided by the IBM Personal Communications family of emulators.

1.4.3 Programming Support

Communications Server and OS/2 Access Feature support a wide range of 32-bit and 16-bit application programming interfaces (APIs) for the application program developer. These APIs provide convenient ways for application programs to access Communications Server functions and allow applications to address the communication needs of connections to both IBM and other computers. In addition, the provided interfaces support SNA protocols so that standardization is ensured. (In this list, Communications Manager refers to both Communications Server and OS/2 Access Feature.)

The APIs supported include:

- Advanced Network Driver Interface Specification (ANDIS)
- Advanced Program-to-Program Communications (APPC)
- Asynchronous Communications Device Interface (ACDI)
- Common Programming Interface for Communications (CPI-C) in both OS/2 and WIN-OS/2
- Communications Manager Kernel
- Communications Manager Common Services Programming Interface
- Communications Manager System Management
- Connection Manager Interface (CMI)

- Conventional LU Application Interface (LUA)
- Generalized Call Control Interface (GCCl)
- IEEE 802.2 (via MPTS)
- NetBIOS (via MPTS)
- Network Driver Interface Specification (NDIS)
- Port Connection Manager (PCM)
- Realtime Interface Co-Processor Device Driver API
- Service Point Application Router (SPA Router)/Remote Operations Service (ROPS)
- Sockets
- X.25

1.4.4 Communication Features

Some of the important communication features provided by Communications Manager include:

- Multiple concurrent hosts support
- Dynamic SNA control point switching
- Multiprotocol Transport Services (MPTS)
- Data security
- SNA data compression
- Asynchronous connectivities
- Synchronous connectivities
- AutoSync connectivities

1.5 Communications Manager for Universal Information Access

Communications Manager functions enable users to access the information they need, when they need it. eNetwork Communications Server for OS/2 Warp Version 5.0 offers a wide spectrum of connectivity options, as well as the flexibility of the multiprotocol capabilities. The mobile user can access the host connection as well, using the SNA Phone Connect feature. With Communications Server, your users experience higher productivity and a more efficient work environment.

The following sections describe:

- SNA Phone Connect
- Connectivity options and services
- SNA over TCP/IP access nodes and gateway
- Sockets over SNA access nodes and gateway
- TN3270E server
- Conformance with SNA

1.5.1 SNA Phone Connect

Use the SNA Phone Connect feature to take advantage of the enhanced wide area network (WAN) connectivity over switched and nonswitched lines. Communications Server and OS/2 Access Feature support automatic dialing capabilities for asynchronous (SNA over Async), synchronous, and AutoSync communications over modems.

SNA Phone Connect uses automatic switched call management: outgoing calls are placed only when the need for a link is detected; incoming calls are automatically answered; and unused links are automatically disconnected. SNA phone connections occur over standard analog public switched telephone networks, digital networks, or an ISDN at speeds up to 64 kbps.

The SNA Phone Connect feature lets you use OS/2 Access Feature from home or while traveling. Your workstation or laptop needs only a modem and the standard serial communications port for asynchronous and AutoSync connections. Your workstation needs a modem and a synchronous adapter for synchronous connections. Supported synchronous adapters include the IBM Multiprotocol Adapter and the IBM Wide Area Connector (WAC) adapter.

SNA Phone Connect supports the SDLC, IDLC, and X.25 protocols.

The SNA functions can use the auto-dial capabilities when you change the connection type of existing configurations to SDLC using SNA Phone Connections in eNetwork Communications Server for OS/2 Warp.

Also, applications written to the following Communications Server APIs can use the auto-dial capabilities without any changes to the application (when you change the connection type of existing configurations to SDLC using SNA Phone Connections in eNetwork Communications Server for OS/2 Warp):

- APPC (LU 6.2)
- CPI-C
- LUA

1.5.2 Connectivity Options and Services

Communications Server and OS/2 Access Feature workstations can communicate using several different types of communications protocols and connection types. (In this discussion, Communications Manager refers to both Communications Server and OS/2 Access Feature.) These include:

- Asynchronous, synchronous, and AutoSync communications using SDLC, IDLC, and X.25 protocols
- IEEE 802.2 over a LAN
- Integrated Services Digital Network (ISDN)
- Synchronous Data Link Control (SDLC)
- General DLC (GDLC)
- X.25
- SNA over TCP/IP (AnyNet)
- Sockets over SNA (AnyNet)

Communications Server and OS/2 Access Feature support a wide variety of communication links. Supported devices include the IBM family of controllers, many IBM protocol converters and bridges, and IBM and other modems. eNetwork Communications Server for OS/2 Warp Version 5.0 workstations can communicate using coaxial, twinaxial, LAN (IBM Token Ring, Ethernet, IBM PC Network, ATM LAN Emulation, Frame Relay and FDDI), switched and nonswitched lines, and ISDN connections.

1.5.3 SNA over TCP/IP Access Nodes and Gateway

The SNA over TCP/IP access node function of Communications Server enables SNA applications to communicate over a TCP/IP network. Application programs, such as Customer Information Control System (CICS) and Information Management System (IMS), can access a TCP/IP network without modification.

The SNA over TCP/IP Gateway function enables SNA applications in SNA and TCP/IP networks to communicate. Using the SNA over TCP/IP Gateway function, SNA applications running non-natively on a TCP/IP network can communicate, without change, with SNA applications on an SNA network. SNA over TCP/IP supports both independent and dependent LU communication:

- For independent LU communication, two or more SNA over TCP/IP gateways can be used to connect multiple TCP/IP networks and SNA networks, thereby allowing SNA applications to communicate with each other across multiple networks.
- For dependent LU communication, the SNA over TCP/IP function can be used with or without the SNA Gateway function to enable downstream workstations in SNA and TCP/IP networks to communicate.

Note: The SNA over TCP/IP Gateway supports 1500 LU-LU sessions.

1.5.4 Sockets over SNA Access Nodes and Gateway

The Sockets over SNA access node function enables C application programs using the IBM TCP/IP AF_INET socket interface to communicate over SNA networks. The Sockets over SNA Gateway function enables socket applications in SNA and TCP/IP networks to communicate. Sockets over SNA gateways can be used to connect remote TCP/IP networks using an SNA backbone network.

1.5.5 TN3270E Server

The TN3270E Server function enables you to configure your network using either a wide area SNA network or a wide area TCP/IP network.

The TN3270E Server function supports:

- LU emulation

TN3270E Server supports LU 2, which enables users to run interactive 3270 display application programs.

- Host print

The Telnet 3270 standard extensions (TN3270E) enable users to print from host applications to printers attached to their workstation or in their TCP/IP network, using LU 1 and LU 3 print sessions.

TN3270E Server implements the protocols outlined in RFC 1646 and RFC 1647, enabling the server to pass LU 1 and LU 3 session data to

TN3270E-enabled clients, wait for client confirmation of the print request, and respond to the host.

- Response handling

TN3270E-enabled clients can send both positive and negative responses, which TN3270E Server sends on to the host. TN3270E Server generates request responses for standard TN3270 clients.

- ATTN and SYSREQ key handling

TN3270E Server can convert and forward information to the host when the client sends an ATTN or SYSREQ key. Clients that are not TN3270E-enabled, however, have no explicit definition for ATTN and SYSREQ.

- LU classes

Communications Server categorizes user connections with LU classes. Classes consist of LUs configured with common characteristics, those that require a specific host connection, for example. This simplifies user access, groups users by application needs, and maximizes host resources.

TN3270E Server supports both standard and extended Telnet 3270. Typical client programs emulate a 3270 display. Clients that support the TN3270E protocol can emulate LU 1 and LU 3 printers.

1.5.6 Conformance with SNA

eNetwork Communications Server for OS/2 Warp Version 5.0 supports conformance with Systems Network Architecture (SNA), the set of conventions and protocols used in IBM networks. SNA helps ensure standardization of network configurations and accurate transmission of data across networks.

An SNA network is organized as a system of nodes and links. Each node is classified according to its capabilities and the extent of control it has over other nodes in the network. The node type is not necessarily associated with a specific type of hardware, and the node's capabilities can be performed by different devices. For example, a workstation acting as a gateway can perform the same functions as a communications controller.

eNetwork Communications Server for OS/2 Warp Version 5.0 supports the following functions:

- CPI Communications and APPC support
- LU support
 - SNA gateway support (Communications Server only)
 - Conventional LU applications (LUA) for LU 0, 1, 2, and 3
- Management services
- System management
- Data Link Control (DLC) definitions

Part 2. Client/Server Programming

Chapter 2. CPI-C for Java

This chapter provides basic information about the installation and configuration of the new CPI-C for Java support included in eNetwork Communications Server for OS/2 Warp Version 5.0. CPI-C for Java allows you to write portable object-oriented APPC client/server applications.

The CPI-C for Java Toolkit implemented in eNetwork Communications Server for OS/2 Warp Version 5.0 provides you with the capability to write truly portable object-oriented client/server applications. The Java run-time environment is required to be available in your system or workstation.

2.1 CPI-C for Java Implementation

Figure 2 shows how the CPI-C class (CPIC.CLASS) has been implemented in the CPI-C for Java option in eNetwork Communications Server for OS/2 Warp Version 5.0.

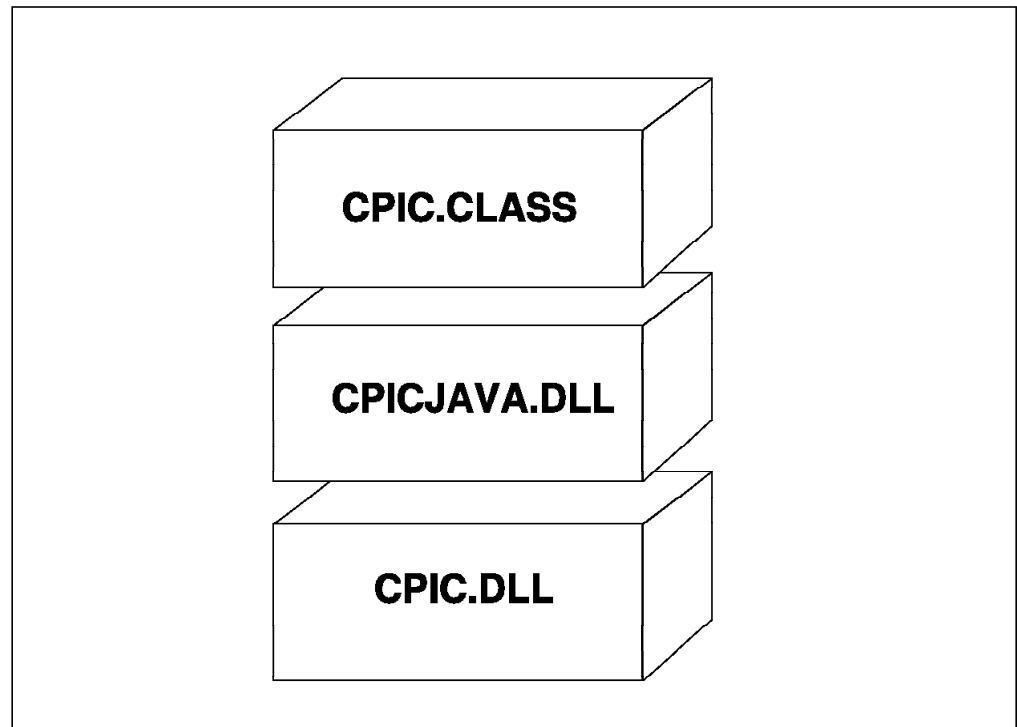


Figure 2. CPI-C for Java Implementation in eNetwork Communications Server for OS/2 Warp Version 5.0

The CPI-C class provided by CPI-C for Java are member functions that mimic the "C" CPI-C API, for example, the CPI-C verbs `cmunit`, `cmaccp`, `cmsend`.

A CPI-C class generates CPI-C objects which are basically CPI-C conversations when methods are executed. Therefore, a CPI-C object can be initialized (`cmunit`), a conversation can be allocated (`cmalloc`), data can be sent (`cmsend`) and so on. The CPI-C class is always associated with CPI-C, for example `CM_CID_SIZE`.

CPI-C for Java provides CPI-C parameter classes also. An instance of these parameters can be generated, for example CPICReturnCode, CPICControlInformationReceived, etc. In the CPI-C for Java implementation, each CPI-C parameter class's constant is an static member data.

2.2 CPICJAVA.DLL

The CPI-C for Java support has not been distributed in the eNetwork Communications Server for OS/2 Warp Version 5.0 CD-ROM, and it has to be downloaded from the networking IBM Web site.

The Web site currently offers support only for OS/2 and it will soon be available for other platforms such as Windows 95 and Windows NT.

Note

There is an old package available for Windows 95 and OS/2 on IBM's Alphaworks Web site. This package is not compatible with the current implementation of CPI-C for Java in eNetwork Communications Server for OS/2 Warp Version 5.0.

2.3 What Is Java ?

Java is a programming language that is object-oriented and it is like C++ but much simpler. Source Java programs are compiled into interpreted bytecodes that can be executed on any machine capable of running a Java virtual machine.

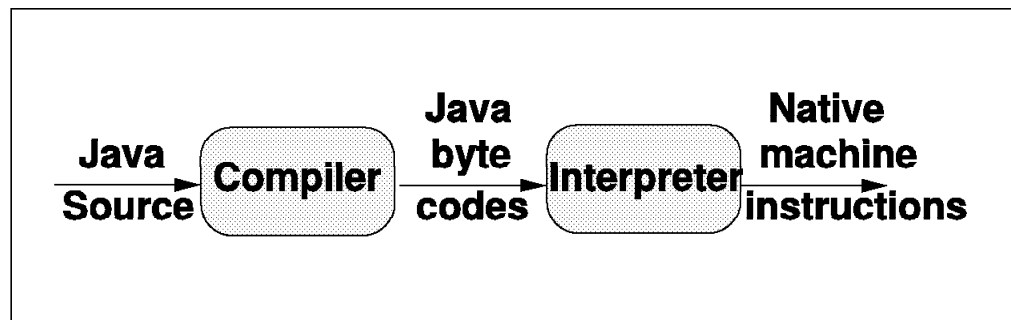


Figure 3. Java Compiler and Interpreter

The Java virtual machine is an interpreter for the bytecodes that makes Java applications platform-neutral and allows dynamically loaded code (see Figure 3).

Note

CPI-C for Java is only implemented for client/server Java applications and not for applets.

2.4 Benefits

There are many benefits gained when you code your APPC client/server applications using CPI-C for Java. For example:

- You write your CPI-C for Java application once and run the application on any platform that supports CPI-C for Java.
- You port the client side applications to CPI-C for Java and run with existing APPC and CPI-C server applications without any change to the server application. In other words, CPI-C for Java client programs communicate with application servers written to other LU 6.2 APIs such as APPC for OS/2, APPC for Windows NT CPI-C, CICS command level and so on.
- CPI-C for Java will leverage existing CPI-C programming skills in the new Java world.

2.5 Installation

In this section we show you how to install the required software in order to support CPI-C for Java client/server applications in your OS/2 system. We include information on how to install the Feature Install (FI), Java for OS/2 1.1.1 and the CPI-C for Java support.

2.5.1 Required Software

In order to be able to run CPI-C for Java applications, the following software is required in your system:

1. Netscape browser for OS/2.
2. Feature Install (FI). You can get the **firunpkg.zip** file from Web site:
<http://finstall.austin.ibm.com/devtk12.htm>
3. Java for OS/2. You can get the **java111.zip** file from Web site:
http://service.boulder.ibm.com/asd-bin/doc/en_us/java111/f-feat.htm
4. CPI-C for Java. You can get the **jcpiccs2.exe** file from Web site:
<http://www.networking.ibm.com/cms/cmscpicj.html>

2.5.2 Installing Feature Install (FI)

You must install FI before attempting any Java installation. The following steps may be followed as a guide to install Feature Install (FI):

1. Create a subdirectory called FISETUP and copy firunpkg.zip to it.
2. Unpack the zip file with PKUNZIP2 firunpkg -d.

Note

Make sure you specify the d option when using PKUNZIP2.

3. Execute **fisetup** to install FI.
4. At this point you should reboot your machine since CONFIG.SYS has been updated.

After properly unzipping the Feature Install (FI), you will find the following files and modules:

The volume label in drive D is D_DRIVE.
The Volume Serial Number is 274C:BC15.
Directory of D:\firunpkg

10-20-97	8:47p	<DIR>	0	.
10-20-97	8:47p	<DIR>	0	..
4-01-97	10:53a	2693	0	botban7.jpg
7-27-97	11:18p	20495	0	cidlog.dll

7-26-97	7:40p	53514	0	fisetup.exe
7-26-97	7:40p	7332	0	fisetup.sym
7-27-97	11:19p	4532	0	genfirpl.sym

7-27-97	11:09p	4276	0	wpinstal.sym
43 file(s)		1501763 bytes used		
		598654976 bytes free		

2.5.3 Installing Java for OS/2 1.1.1

In this section we describe how Java for OS/2 1.1.1 is installed. The Java Toolkit will allow you to compile your client/server programs and the Java run-time environment will allow you to execute your programs under the Java virtual machine. You may want to execute the following steps in order to have the Java Toolkit (JDK) installed in your machine:

1. Create a subdirectory called java111 and copy java111.zip to it.
2. Unpack the zip file with PKUNZIP2 java111 -d.

Note

Make sure you specify the d option when unpacking.

3. Execute **Install** and follow the directions. There are two ways you can install Java 1.1.1:
 - Guided Path
 - Advanced Path, which allows you to select target drives and target subdirectories.

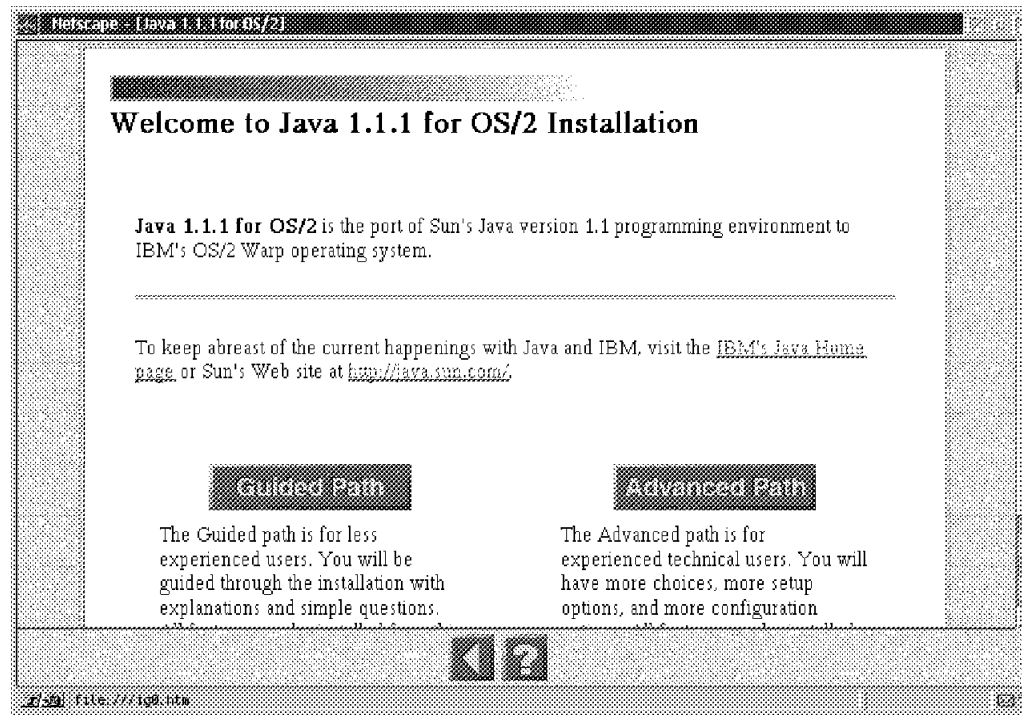


Figure 4. Java 1.1.1 Installation

Figure 4 illustrates the Java 1.1.1 installation and options.

When using the Guided Path you will be prompted to select the components to be installed (see Figure 5).

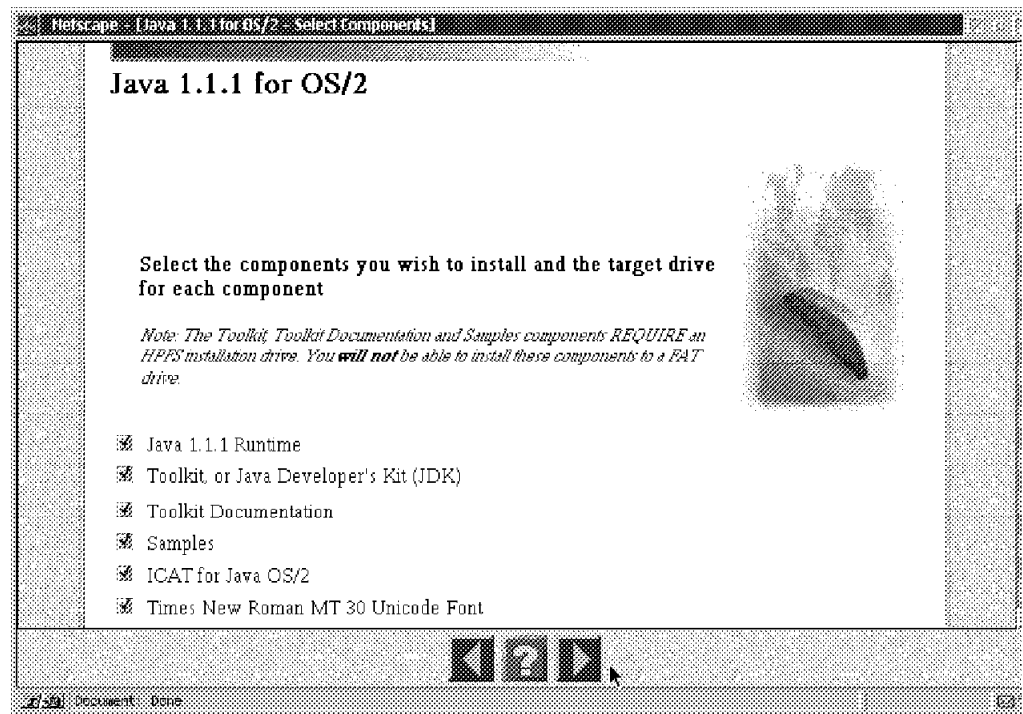


Figure 5. Java 1.1.1 Guided Path Installation

When using the Advanced Path you will be prompted to select the components to be installed as well as the target subdirectories (see Figure 6 on page 24).

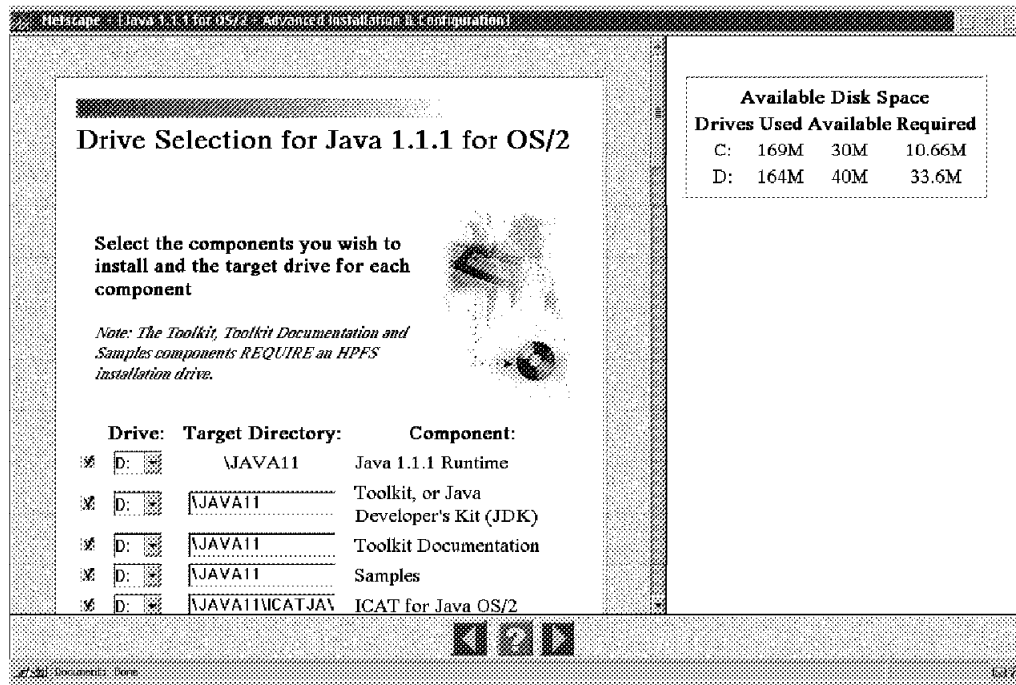


Figure 6. Java 1.1.1 Advanced Path Installation

After a successful installation of Java 1.1.1, you should find the following entries in the target subdirectory:

The volume label in drive D is D_DRIVE.
The Volume Serial Number is 274C:BC15.
Directory of D:\java11

```

10-20-97  9:00p  <DIR>          0  .
10-20-97  9:00p  <DIR>          0  ..
10-20-97  9:00p  <DIR>          0  bin
  7-14-97  6:51a   3542      0  cnfgappl.htm
  7-26-97 12:30a    499      0  copyrght
10-20-97  9:00p  <DIR>          0  demo
10-20-97  9:00p  <DIR>          0  dll
10-20-97  9:01p  <DIR>          0  docs
  7-26-97 12:31a   4394      0  folder1.ico
  7-26-97 12:31a   4394      0  folder2.ico
10-20-97  9:01p  <DIR>          0  ICATJAVA
10-20-97  9:01p  <DIR>          0  include
  7-15-97 10:19a   2548      0  index.html
  7-26-97 12:31a   7194      0  javacmpt.bmp
10-20-97  9:00p  <DIR>          0  lib
  7-30-97 10:46a  47938      0  readme
  7-15-97 10:19a   9781      0  runtime.htm
  7-15-97 10:19a  22126      0  toolkit.html
10-20-97  9:01p  <DIR>          0  uninstal
      19 file(s)      102416 bytes used
                        598654976 bytes free

```


2.5.4 Installing CPI-C for Java

In this section we show how to install the CPI-C for Java support in eNetwork Communications Server for OS/2 Warp Version 5.0. You may want to take the following steps in order to install CPI-C for Java:

1. Create a subdirectory (for example, JCPIC) and copy the downloaded jcpiccs2.exe file.
2. In this subdirectory, enter: **jcpiccs2** to expand the CPI-C for Java modules.

After a successful installation of CPI-C for Java, you should find the following entries in the subdirectory:

The volume label in drive D is D_DRIVE.

The Volume Serial Number is 274C:BC15.

Directory of D:\jcpic

10-16-97	4:37p	<DIR>	0	.
10-16-97	4:37p	<DIR>	0	..
9-08-97	2:13p	14336	49	CPICJAVA.dll
9-09-97	9:34a	52159	0	cs2cpicj.jar
9-09-97	8:55a	<DIR>	0	docs
10-10-97	8:07a	507550	42	jcpiccs2.exe
9-09-97	9:05a	204	0	jcsetup.cmd
9-09-97	8:40a	<DIR>	0	Samples
8 file(s)			574249 bytes used	
			598653440 bytes free	

Note

The **cs2cpicj.jar** file contains CPI-C for Java classes.

The Samples subdirectory contains the JPing.java program:

The volume label in drive D is D_DRIVE.

The Volume Serial Number is 274C:BC15.

Directory of D:\jcpic\samples

10-16-97	4:37p	<DIR>	0	.
10-16-97	4:37p	<DIR>	0	..
10-30-97	12:15p	7154	0	JPing.class
10-23-97	2:38p	16263	35	JPing.java

Note

The JPing.class file was created after the JPing.java sample program was compiled.

2.6 Compiling and Executing Sample Java Programs

In this section we show how to compile and execute Java programs so you can become familiar with the process.

When you install Java, you also install the Java sample program, for example **Hello**.

The "hello world" sample program is in the \demo\hello subdirectory in the Java 1.1.1 directory. You can execute this program with the command: **java hello**.

If you would like to change the "hello world" message, you can do something like this:

1. Erase the file hello.class.
2. Edit the file hello.java and change the parameter `System.out.println("Hello World")` to say "Hello your name" for example.
3. Compile the hello.java program with the command:
`javac hello.java`
4. Execute the `dir` command in hello subdirectory, and verify that there is a new hello.class program.
5. Execute the new hello.java program.

2.7 Compiling and Executing CPI-C for Java Programs

In this section we show how to compile and execute CPI-C for Java programs. In the JCPIC directory you will find the Samples subdirectory. The CPI-C for Java support provides a JPing sample program (client application only).

In the JCPIC directory, note that the **cs2cpicj.jar** contains the required classes to compile a CPI-C program using Java. For example, the `cpic.class` is in this jar file.

You can edit the `jpings.java` program and see how a CPI-C object is generated. For example:

- `public static CPIC cpicObject = new CPIC();`

You can also see how the CPI-C class is used with CPI-C messages (CPI-C verbs) such as:

- Initialize conversation: `cpicObject.cminit(..parameters..)`
- Set maximum buffer size: `cpicObject.cmembfs(..parameters..)`
- Allocate conversation: `cpicObject.cmallc(..parameters..)`
- Set send type: `cpicObject.cmsst(..parameters..)`
- Send data: `cpicObject.cmsend(..parameters..)`
- Receive data: `cpicObject.cmrcv(..parameters..)`
- Set deallocate type: `cpicObject.cmsdt(..parameters..)`
- Deallocate conversation: `cpicObject.cmdeal(..parameters..)`

To compile the `jpings.java` program, enter the command: **javac jpings.java**

2.7.1 Troubleshooting

Here we describe some possible errors you may run into when executing a CPI-C for Java program:

- If you receive the message: Unable to load dll 'CPICJAVA' (errcode = 'CPICJAVA'), you probably do not have the CPICJAVA.DLL in the libpath of the config.sys. You should probably copy the CPICJAVA.DLL to the \cmlib\dll\

subdirectory. You can also do this by executing the provided jcsetup.cmd batch file.

- If the compile failed indicating that import COM.ibm.eNetwork.cpic.*; could not be found, you probably do not have the cs2cpicj.jar file in the CLASSPATH environment variable in the config.sys (c:\jcpic\cs2cpicj.jar).
- If the compile failed indicating that import java.io.*; could not be found, you probably do not have the classes.zip file in the CLASSPATH environment variable in the config.sys (c:\java11\lib\classes.zip).

2.7.2 Running JPing

JPing is a client CPI-C sample program and it only takes a symbolic destination name for a partner LU, mode and TP name. Therefore you are required to create a CPI-C Side Information entry in your SNA configuration.

Note

You cannot use a fully-qualified partner LU in this version of JPing.

You may want to execute the following steps to create a CPI-C Side Information entry in your configuration:

1. Use CMSETUP to enter the configuration process.
2. Select **SNA Features**.
3. Select **CPI-C Communications Side Information**, as shown in Figure 7 on page 28.
4. Select **Create**.
5. Enter a symbolic name, for example TOVTAM (uppercase).
6. Enter partner LU fully qualified name that you want to JPing, for example USIBMRA.RAK (VTAM). The partner LU fully qualified name is not case sensitive.
7. Enter the application server TP name as APINGD (uppercase). Note that TP names are always case sensitive.
8. Select mode #INTER (uses COS with high priority).
9. Verify and save your configuration.

Figure 7. CPI-C Communications Side Information

2.7.2.1 Executing JPing

To execute the JPing program you will enter:

```
java jping -s TOVTAM
```

Note

Make sure you have Communications Server (or OS/2 Access Feature in client workstations) running!

In a successful installation, JPing will execute and display the results, for example:

```
-----JPing Opening-----
      OS Name      = OS/2
CPI-C Side Info (-s) = TOVTAM
Number Iterations (-n) = 20
      Data Length (-d) = 2000
      Max Buffer (cmembs) = 65535

Allocate Time = 0.0030 seconds

Version Exchange Time = 0.163 seconds
```

```

Server Error Version: 1
ServerVersion: 2.33

Ping Time = 0.0050 seconds
Ping Time = 0.0050 seconds
Ping Time = 0.0050 seconds
Ping Time = 0.0050 seconds
Ping Time = 0.0060 seconds
-----
Ping Time = 0.0040 seconds
Ping Time = 0.0030 seconds
Ping Time = 0.0030 seconds
Ping Time = 0.0030 seconds
Ping Time = 0.0040 seconds
Ping Time = 0.0030 seconds

Total Ping Time = 0.092
Average Ping Time = 0.0046
-----JPing Closing-----

```

2.7.3 Diagnosing CPI-C Errors

CPI-C errors are handled just like any other "C" CPI-C errors. For example:

- Use CPI-C reference
- Capture trace
- Check configuration: side information, TP definitions, and security
- Verify attach manager is started as required.

2.8 Implemented CPI-C Calls on OS/2

All CPI-C calls supported by Communications Server for OS/2 Warp are supported except non-blocking calls:

- cmwcmp: Wait for Completion
- cmsqcf: Set Queue Callback Function
- cmsqpm: Set Queue Processing Mode
- cmspm: Set Processing Mode
- cmwait: Wait for Conversation

Note

CPI-C for Java does not support non-blocking calls in eNetwork Communications Server for OS/2 Warp Version 5.0.

Part 3. Client Workstations

Chapter 3. Windows 95/NT Client Workstations

The Windows NT Access Feature provided by eNetwork Communications Server for OS/2 Warp enables applications that are written to the Sockets, APPC, CPI Communications, and LUA APIs to run unchanged over SNA and TCP/IP local and wide area networks. In this chapter we present an overview of the Windows NT Access Feature and describe the installation process.

3.1 Overview

The Windows NT Access Feature in eNetwork Communications Server for OS/2 Warp provides communications to host machines and other connections from Windows 95 or Windows NT client workstations.

This feature includes support for Advanced-Peer-to-Peer Networks (APPN) as an end node (EN), and uses the advanced network features: high performance routing (HPR) and dependent LU requester (DLUR). It also includes AnyNet SNA over TCP/IP, which allows client/server SNA applications to communicate over a TCP/IP network and it can be installed on a workstation using Windows 95 or Windows NT.

The sections in this chapter provide information on how to install these options using the Communications Server CD, a code server, diskettes, or a response file reference.

3.2 Installing the Windows Access Feature from CD

To install the Windows Access Feature using the Communications Server CD, follow these steps:

1. Insert the CD into the CD-ROM drive.
2. From an MS-DOS session change directories to the CD-ROM drive.
3. Type `install`, or select **Start**, then **Run** from your Windows 95 or Windows NT session.
4. When the Install Program is displayed, click on the Installation icon.
5. When the Installation window is displayed, select the Windows 95 and Windows NT Access Feature radio button.
6. Select the **Begin** button.

3.3 Installing the Access Feature Using Code Server

To install the Windows 95/NT Access Feature using a code server, follow these steps:

1. Run the commands provided by your administrator to attach to the code server. These commands can include attaching to a LAN drive and changing to the subdirectory where the access feature files are located (for example `x:\clients\win32`).
2. Enter `win install` at the DOS prompt.

3. Select one of the products that you want to install on the Installation and Maintenance window.
4. Make the appropriate selections on the Install window and click on **OK**.
5. Select the components that you want to install on the Install - directories window and click on **Install**.
6. When you have installed all of the products that you want, exit the installation utility by pressing F3 twice or by closing the installation program windows.
7. Shut down and restart your workstation.

3.4 Installing the Access Feature Using Diskettes

To install the Windows 95/NT Access Feature using diskettes, follow these steps:

1. Insert the first diskette in the diskette drive.
For information on how to create diskettes see 3.4.1, "Creating Diskettes."
2. Select **Start** and then click on **Run**.
3. In the Run window, type `a:install` on the command line.
4. Click on **OK**.
5. Make the appropriate selections on the Install window and click on **OK**.
6. Select the components that you want to install on the Install - directories window and click on **Install**.
7. Exit the installation utility by pressing F3 twice or by closing the installation program windows.
8. Shut down and restart your workstation.

3.4.1 Creating Diskettes

Diskette images for the access features are stored in platform-specific subdirectories on the CD-ROM.

Be sure to make copies of all products needed to install a particular access feature. The following software prerequisites are mandatory:

- OS/2 Access Feature requires that MPTS (shipped on the CD-ROM) be installed and configured.
- APPC Networking Services for Windows requires the IBM Local Area Network Support program (shipped on the CD-ROM).
- AnyNet APPC over TCP/IP for Windows requires APPC Networking Services and one of the TCP/IP products listed under AnyNet APPC over TCP/IP for Windows.

To create diskettes to install one or more of the access features, follow these steps:

1. Open an OS/2 window. If you are creating diskettes from the CD-ROM drive in drive E:, change to the directory that contains the diskette images that you want, for example:
`e:\diskimgs\clients\xxxxxxx`

where xxxxxxx is: OS2\WARPAF, OS2\OS2211AF, WINDOWS\ANYNET, WINDOWS\NETSUPP, or WINDOWS\NSW.

If you are creating diskettes from another drive, replace the E: with that drive.

2. Determine the number of diskettes needed by entering `dir` at the command prompt.
3. Use the `LOADDSKF` command to load a copy of the diskette images onto diskettes. For example, to load the first diskette for APPC Networking Services, enter:

```
e:\diskimgs loadskf nswinst1.dsk a: /f
```

The `/F` indicates that the diskette will be formatted to match the input image.

`LOADDSKF` is located in the `\DISKIMGS` subdirectory. To get help on this command, enter `loadskf` at the command prompt.

3.5 Installing Access Feature Using a Response File

You can install the Windows 95/NT Access Feature using a response file, but you cannot configure it with a response file.

To install Windows 95/NT Access Feature using a response file, follow these steps:

1. Run the commands provided by your administrator to attach to the code server. These commands can include attaching to a LAN drive and changing to the subdirectory where the access feature files are located.
2. Select **Start** and click on **Run....**
3. In the Run window, enter:

```
x:\clients\win32\install.exe /a:I /l1:installation_error_log_file  
/l2:installation_history_log_file  
/p:"Networking Services for Windows 95/NT" /r:response_file  
/s:x:\clients\win32 /x
```

`/A:I` is an action that indicates that the product is being installed for the first time.

`/L1:installation_error_log_file` specifies the fully qualified path and name of the installation error log file. This parameter is not required.

`/L2:installation_history_log_file` specifies the name of the history file. This parameter is not required.

The `/P` parameter specifies the product name.

`/R:response_file` specifies the response file name (.RSP).

The `/S` parameter specifies the directory that contains the source files to be installed. `X:\CLIENTS\WIN32` indicates the LAN-accessed directory where the Windows 95/NT access node files are located. If you are installing from another drive, replace `X:` with that drive.

The `/X` parameter suppresses installation windows.

Refer to the Response File Reference for more information on CID installation.

3.6 SNA Node Configuration

We will follow the Configuration of an SNA end node for the node scenarios. First we need to configure a node. For our example we will call the node WSWIN95 in the Windows 95 client and WSNT in the Windows NT client. You should add the fully qualified CP name as shown in Figure 8 on page 37, and the CP alias too.

The Access Feature Node Configuration provides configuration options that enable you to define SNA nodes and associated resources.

When creating a new SNA configuration (a new node and associated resources), use the following configuration options:

- Configure Node
- Configure Devices
- Configure Connections
- Configure Modes
- Configure DLUR PUs
- Configure Local LU 6.2
- Configure Partner LU 6.2
- Configure Local LU 0 to 3
- Configure CPI-C Side Information
- Configure Transaction Programs
- Configure Connection Networks
- Configure Focal Points
- Configure User ID Passwords
- Configure LU-LU Passwords
- Configure Sockets over SNA

To define a new resource, click **New**.

To display, change, or add a definition in the list for a resource:

1. Select the configuration option.
2. Select the definition.
3. Click **View/Change/Add**.

The resource must be inactive before redefining. You can use SNA Node Operations to see if the resource is active.

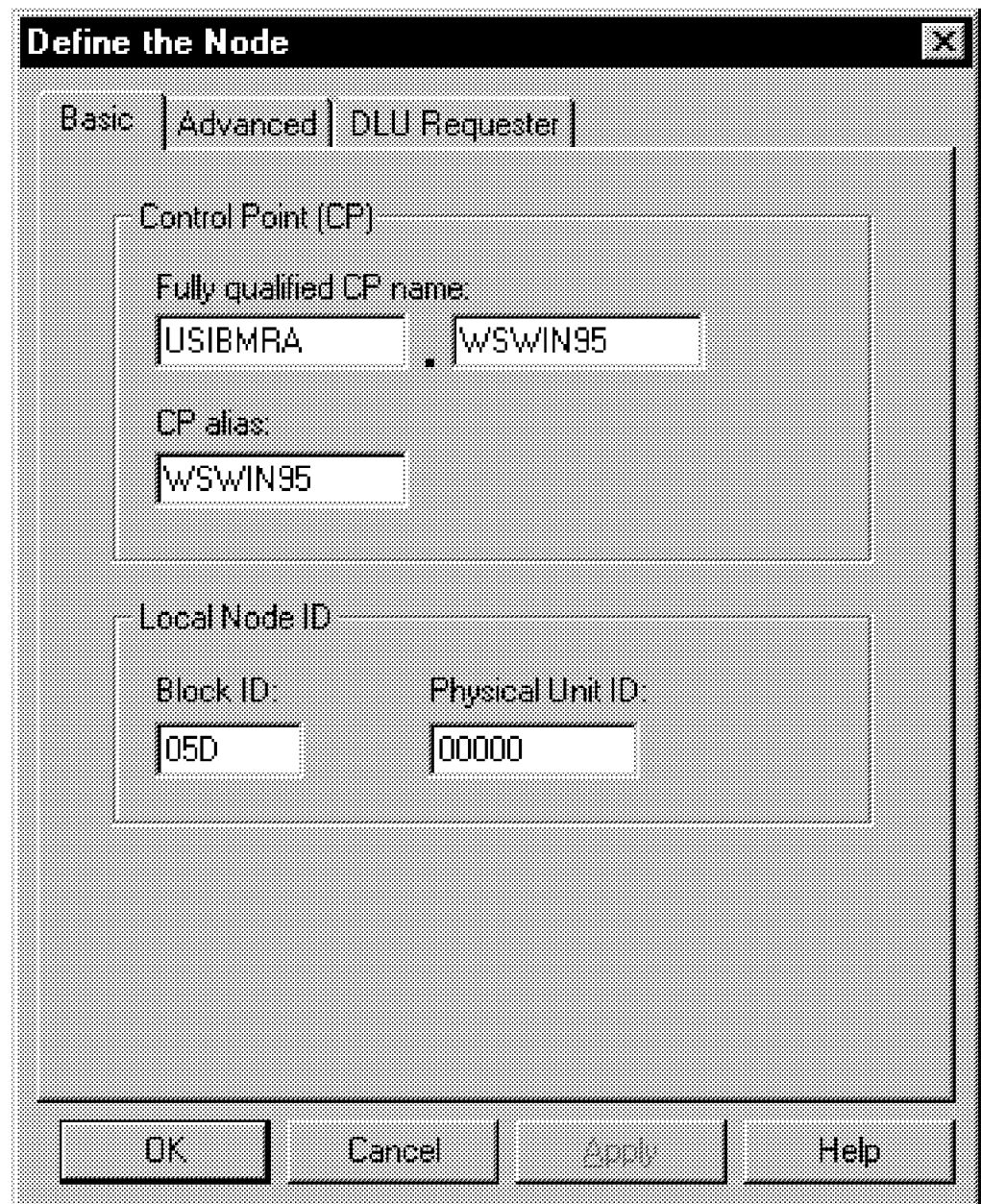
Changing the name of the definition creates a new definition. Changing other parameters alters the original definition (and does not create a new definition).

To delete a definition in the list for a resource:

- Select the configuration option.
- Select the definition.
- Click **Delete**.

3.6.1 Define Node

The Systems Network Architecture (SNA) local node characteristics contain the parameters needed to identify your workstation to the SNA network. A host that is performing network management needs the network ID and the control point (CP) name, also known as the local node name, so that when the workstation sends alerts to the host, the host recognizes the application running on the workstation.



The screenshot shows a Windows-style dialog box titled "Define the Node" with a close button (X) in the top right corner. The dialog has three tabs: "Basic", "Advanced", and "DLU Requester". The "Basic" tab is selected. Inside the dialog, there are two main sections. The first section is labeled "Control Point (CP)" and contains two text input fields. The first field is labeled "Fully qualified CP name:" and contains the text "USIBMRA" followed by a period "." and the text "WSWIN95". The second field is labeled "CP alias:" and contains the text "WSWIN95". The second section is labeled "Local Node ID" and contains two text input fields. The first field is labeled "Block ID:" and contains the text "05D". The second field is labeled "Physical Unit ID:" and contains the text "00000". At the bottom of the dialog, there are four buttons: "OK", "Cancel", "Apply", and "Help".

Figure 8. Define Node - Basic

For the node definition, there are options for the local node alias name, local node ID, and node type. You can also configure how the node registers its resources in an APPN network and how it discovers (or serves) LAN addresses for configuring new LAN links into the network.

Through the DLUR characteristics, you define the primary and secondary dependent LU servers (DLUS) for serving dependent LU sessions across an APPN network to the local node.

Use Configure Node to define the local node. Resources can be associated with the local node.

To define a new node:

1. Select **Configure Node**.
2. Click **New**.
3. Use the Basic fields to define the basic parameters. The basic parameters are required. See Figure 8 on page 37.
4. Click the Advanced tab to define the advanced parameters. Defining the advanced parameters is optional. If you do not define these parameters, the defaults are used. Here you specify the APPN options to allow your workstation to register its resources with a network node server and with a central directory server. See Figure 9 on page 39.

Note

VTAM is the only platform that supports a central directory server.

5. Click the DLU Requester tab to define the DLU requester default parameters. Defining the DLU requester default parameters is optional. If you do not define these parameters, you can still define some DLUR parameters such as the DLUS name when you define a DLUR PU for PU concentration or DLUR passthru.

To cancel at any time, click **Cancel**.

To display, change, or add the definition of the node:

1. Select **Configure Node**.
2. Click **View/Change/Add**.
3. Select one of the following options:
 - Basic
 - Advanced
 - DLU Requester

To cancel at any time, click **Cancel**.

To delete the definition of the node:

1. Select **Configure Node**.
2. Click **Delete**.

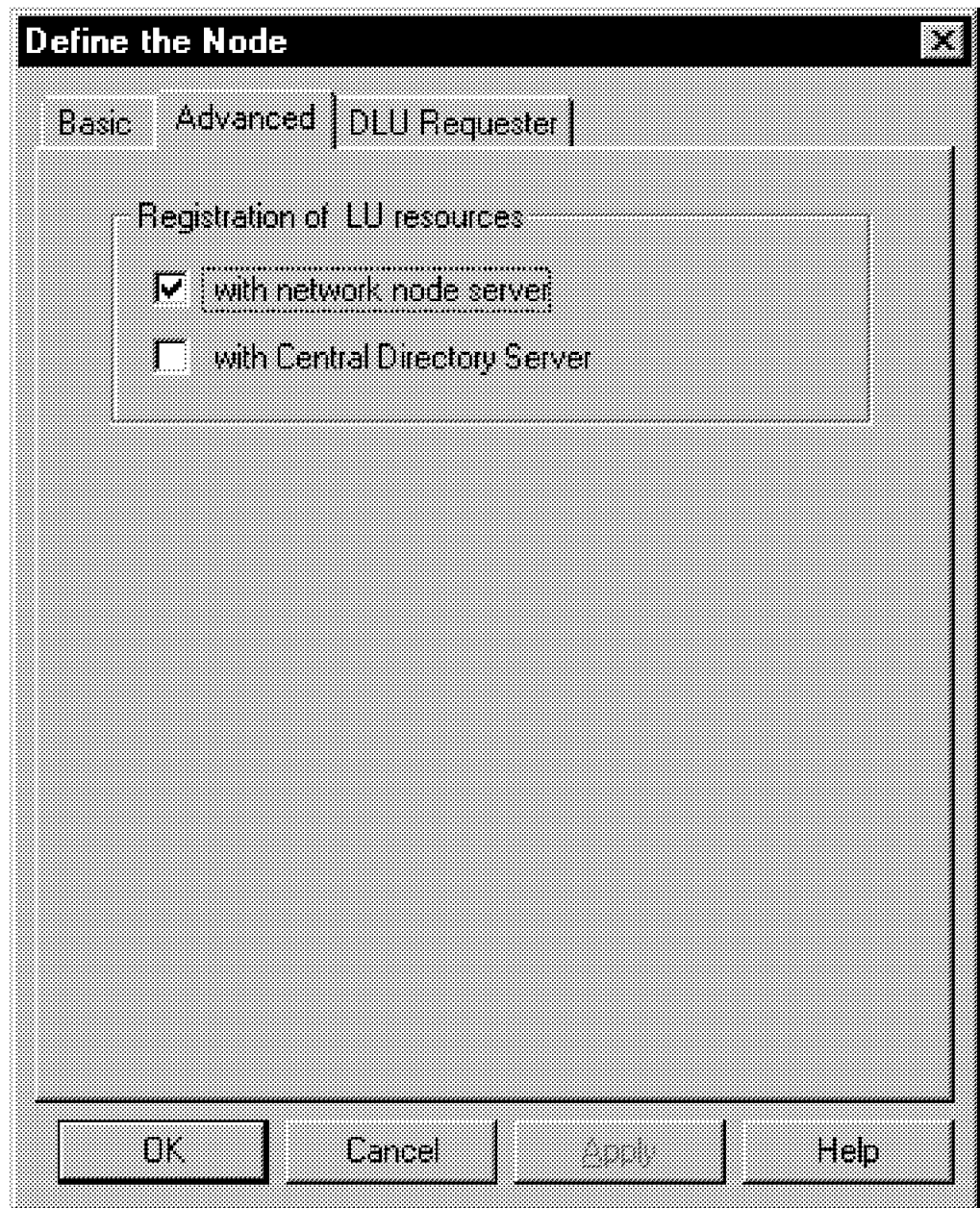


Figure 9. Define Node - Advanced

3.6.2 Configure Devices

A device definition establishes an interface over the DLC API between the adapter in your workstation and another system or workstation. This interface or port is used to support inbound connections and individually defined outbound connections. The DLC provides the protocol necessary for reliable delivery of basic transmission units (BTUs) between a pair of nodes in the SNA network. You must configure the appropriate Personal Communications device definition for a workstation to access an SNA network.

Use Configure Devices to define a new port or modify or delete an existing port.

To configure a device:

1. Select **Configure Devices**.
2. Select one of the data link controls (DLCs):
 - LAN
 - COM Port
 - SDLC-MPA
 - SDLC-WAC
 - Twinaxial
 - AnyNet SNA/IP
 - X.25-ComPort
 - X.25-WAC
 - OEM-DLC (The OEM-DLC selection is displayed only when the OEM device is installed.)

Therefore in our scenario we need to configure the DLC device. In our case we are using token-ring LAN0_04 for adapter number 0 and SAP 4. See Figure 10.

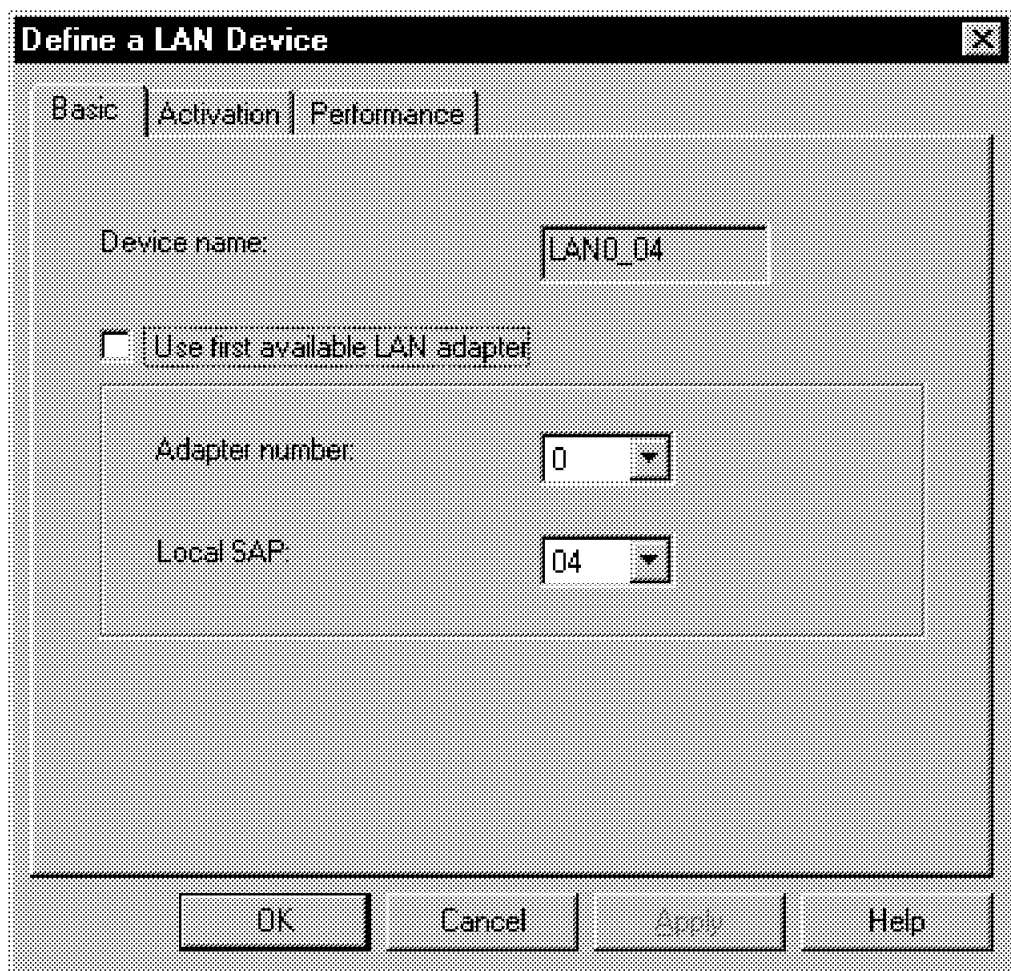


Figure 10. Device Configuration - LAN

3.6.3 Define Connection

Once a device has been configured, you will need to define logical links or connections to others node if required. The exception is when your workstation is an end node (EN) and needs to connect to a network node (NN). In this case, if both nodes are enabled for discovery, the client station will try to dynamically establish an APPN connection to the server.

To define a new LAN connection:

1. Select **Configure Connections**.
2. Select the LAN data link control (DLC).
3. Click **New**.
4. Use the Basic fields to define the basic parameters. The basic parameters are required. See Figure 11 on page 42.
5. Use the Advanced fields to define the link parameters. The advanced parameters are required. See Figure 12 on page 43.
6. Click the **Adjacent Node** tab to define the security parameters. Defining the security parameters is optional. If you do not define these parameters, the defaults are used.

To display, change, or add the definition of a LAN connection in the list:

1. Select Configure Connections.
2. Select the LAN data link control (DLC).
3. Select the definition in the list.
4. Click **View/Change/Add**.
5. Select one of the following options:
 - Basic
 - Advanced
 - Adjacent Node

Define a LAN Connection

Basic | Advanced | Adjacent Node

Link station name: TOSRVB

Device name: LANX_04

Discover network addresses...

Destination address: 400052005198

Remote SAP: 04

☒ Token-Ring ☐ Ethernet

OK Cancel Apply Help

Figure 11. Destination MAC Address

The Basic parameters are:

- | | |
|----------------------------|---|
| Link station name | This name is used to identify the connection when monitoring or event logging. If your workstation requires dependent LU sessions (for example, 3270 sessions) and it does not use a DLUR PU in your workstation, this name will also be the name of the PU. For example, your workstation may be directly connected to a host system or to an SNA gateway in a server machine. |
| Device name | The name of a previously defined device. |
| Destination address | For a LAN connection this is the destination or target address your workstation is connected to. You may enter this address manually or you may want to use the discovery function if both the client (your workstation) and the server are enabled to search and listen for a MAC address. |
| Remote SAP | This is the service application point number used by the remote node for SNA over LAN traffic (IEEE 802.2). You will normally not change the default value of 4 unless it conflicts with another IEEE 802.2 application. |

LAN address format If you are using an Ethernet adapter you may want to configure the option to use the MAC address in canonical form or not.

A very nice feature is the Discover Network Address feature included in this client, so you do not need to remember the MAC address of your server. See Figure 11 on page 42. If the network is big enough several MAC addresses will be available, so be sure to select the correct one.

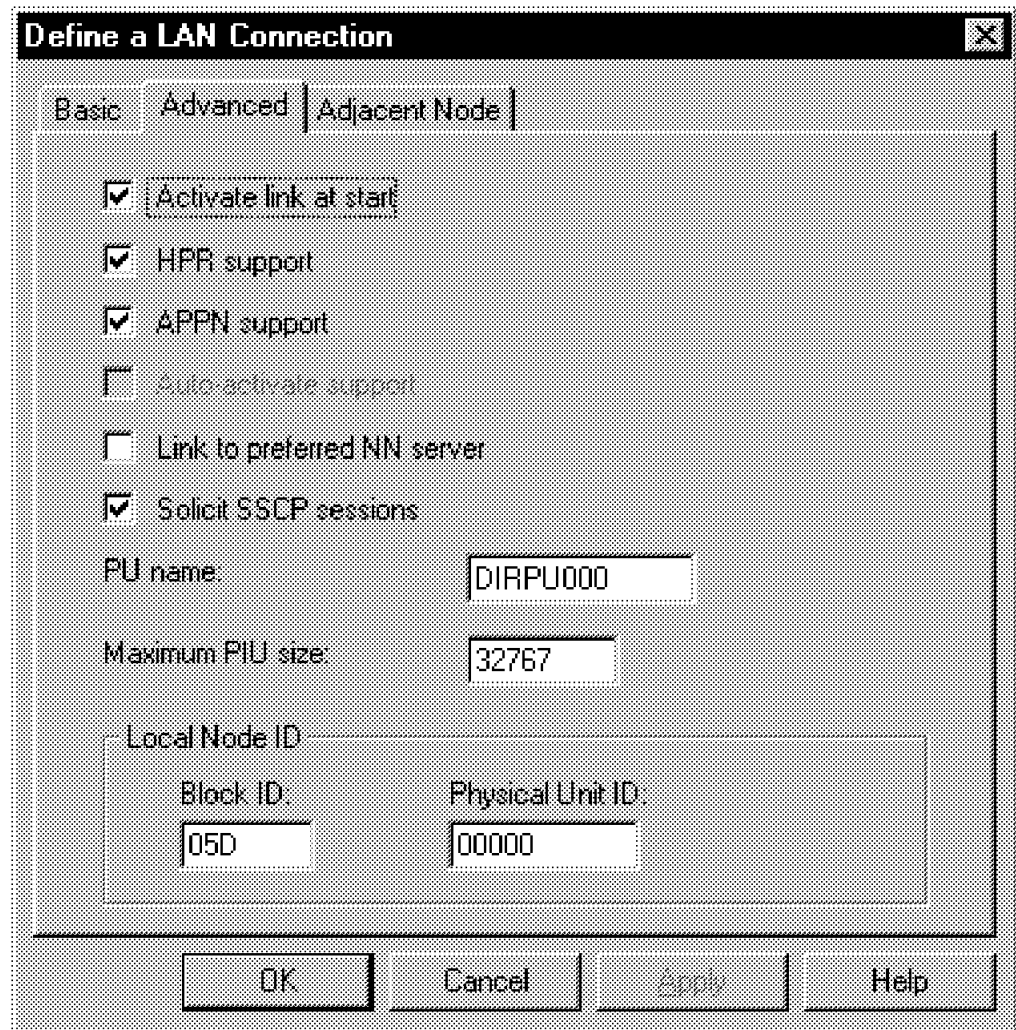


Figure 12. Define a LAN Connection

The Advanced parameters are:

Activate link at start If you do not select this option you can still activate the link either manually or from the remote system you are connected to.

HPR support Select this option if you want your node to negotiate the use of HPR in the XID3 exchange. If the remote node is also enabled for HPR then the two nodes will negotiate the HPR architecture options.

APPN support You will normally be required to enable APPN unless you only have dependent LU sessions (such as 3270 emulation) directly connected to a VTAM subarea or to an SNA

gateway. You do not need APPN support if you are running SNA over TCP/IP, since AnyNet SNA over TCP/IP connections are not APPN connections and behave more like LEN connections.

Auto-activate support

Select this option when you need to dynamically activate this connection upon request.

Link to preferred NN server

If you are connected to multiple network nodes you have the option to configure a preferred NN server in order to establish CP-CP sessions.

Solicit SSCP sessions

This option must be selected when you have dependent LU sessions (for example, 3270) and you are not using a DLUR PU in your workstation. In this case, the connection can be directly to a VTAM subarea system or to an SNA gateway.

Note

Selecting this option causes your node to request an ACTPU (activate PU) in the XID3 when the connection is established.

PU name

This is the name that will be used to identify the PU for this connection.

Block ID

It is part of the node ID and the default value is 05D.

Physical Unit ID

This value is also known as IDNUM and it is part of the node ID.

Note

In an APPN network, the node ID is not used unless the adjacent node is a VTAM system. If you are defining a connection to a VTAM system, you should always check with your VTAM administrator for a proper and valid node ID (block ID/PU ID) that you can use in your node.

Maximum PIU Size

It specifies the maximum packet size for this connection. If larger messages are sent they will be fragmented. PU 2.1 will negotiate this value with the adjacent node and during XID3 exchange when the connection is established.

In our sample configuration, we select HPR support for our APPC sessions and solicit SSCP sessions to access the SNA Gateway for our 3270 emulation sessions.

Note

If you do not configure a DLUR PU in your workstation, your dependent LU session traffic will not go over HPR connections.

Also, if you do not have a DLUR PU in your workstation, you need to solicit SSCP sessions (ACTPU SNA command) from the adjacent node you are connected to. The adjacent node can be either a VTAM/NCP system (PU 4/5) or an SNA gateway (PU 4).

3.6.4 Defining a DLUR PU

If you have dependent LU sessions in your workstation (such as 3270 sessions) and your workstation is connected to an APPN network, you can configure one or multiple DLUR PUs to support your sessions and reach a Dependent LU Server (DLUS) in a VTAM system.

The screenshot shows a Windows-style dialog box titled "Define a DLUR PU". It has a "Basic" tab selected. The fields are as follows:

- PU name: WSWIN95
- Block ID: 05D
- Physical Unit ID: 00000
- Auto-start: ☐
- Use global defaults: ☐
- DLUS name: USIBMRA
- DLUS name: RAK
- Backup DLUS name: (empty)
- Buttons: OK, Cancel, Apply, Help

Figure 13. Defining a DLUR PU

The DLUR PU profile defines the connection between a workstation and a dependent LU server (DLUS). The DLUS handles communications between dependent LUs through an APPN network to a dedicated host.

This profile lets you define:

Physical Unit (PU) name

This name will be used when defining your LUs for this PU. It is also used for event logging.

Auto-start

You will normally select this option to have the DLUR/DLUS connection initiated when you start communications in your node. If you did not select this option you can still initiate the connection manually or let the DLUS initiate it.

Node ID

This is the node ID (block ID and PU ID) of the DLUR PU. It must be unique and either statically or dynamically defined in your VTAM DLUS system.

Note

Always check with your VTAM coordinator for a proper and valid node ID that you can use for your DLUR PU.

Use global defaults

Allows you to use DLUR default values previously defined in the Define Node profile.

Name of the dependent LU server (DLUS)

If you did not configure the default value for the DLUS server name, you must enter the fully-qualified name of the DLUS at this time. The DLUS name is the fully-qualified name of your VTAM system.

Backup DLUS name

Enter a backup DLUS name if you have one, otherwise leave blank.

The specified DLUS must be reachable through the APPN network. If you cannot establish a connection, you may want to APING your VTAM system to make sure you have connectivity.

Note

If you define a DLUR PU in your workstation, the SNA gateway in the server is not used and therefore PU concentration will not be available for the workstation. PU concentration is a function in the SNA gateway that allows you to save in the number of PUs defined in the host system in order to support a large number of workstations.

3.6.5 Defining Dependent LUs

In this section we show you how to configure local dependent LUs (LU 0 to 3) to be used by user applications (LU0) or emulation sessions (LU1, LU2 and LU3). The logical units (LUs) are associated to a physical unit (PU). The PU can be either a subarea PU when using subarea networks or a DLUR PU when connecting to an APPN network.

When using a subarea network, the logical link or connection becomes the subarea PU. However, when connecting to an APPN network you will need to configure a DLUR PU as explained in 3.6.4, "Defining a DLUR PU" on page 45.

The screenshot shows a Windows-style dialog box titled "Define a Local LU 0 to 3". It has a "Basic" tab selected. The dialog contains five labeled fields, each with a text input or dropdown menu:

- LU name:** A text box containing "MYLU3270".
- NAU address:** A dropdown menu showing "2".
- PU name:** A dropdown menu showing "DIRPU000".
- LU model:** A dropdown menu showing "3270-2".
- LU priority:** A dropdown menu showing "Medium".

At the bottom of the dialog are four buttons: "OK", "Cancel", "Apply", and "Help".

Figure 14. Define a Local LU0 to 3 (LU-A)

The LU 0 to 3 definitions provide the parameters that are needed to establish SNA LU0, LU1, LU2, and LU3 communication sessions between a workstation and a host application.

LU-A requires one of the following target host systems:

- S/390 architecture host
- AS/400

The following options or default values are configured:

LU name	This is the name to identify the logical unit (LU). It is only used locally and does not have to match the label of the LU statement in your VTAM system.
NAU address	This is the network addressable unit also known as the local address of the dependent LU. This value has to match the corresponding LU local address in your VTAM system.
PU name	This is the name of a predefined subarea PU or DLUR PU with which this LU is associated.

LU model	Specifies the LU model for dynamic definitions of LUs in your VTAM system.
LU priority	Defines the SNA priority for the session data traffic when using this LU. It can be high, medium or low priority. Default value is medium priority.

3.7 Configuring PCOMM Emulation Sessions

In this section we show you how to configure PCOMM in order to access an OS/2 Communications Server machine. We describe a telnet 3270 connection to access the TN3270E server and a 3270 over LAN connection to access the SNA gateway.

Note

PCOMM emulation is not included in eNetwork Communications Server for OS/2 Warp for client workstations.

3.7.1 Using Telnet 3270

In this section we show you the configuration of a Windows NT client workstation in order to establish 3270 sessions. A telnet 3270 emulation session uses TCP/IP to access a TN3270E server. For details on how to configure a TN3270E server refer to Chapter 10, "TN3270E Server in Branch Extender Nodes" on page 213.

When selecting the communications option in PCOMM, you enter the Customize Communication panel (see Figure 15).

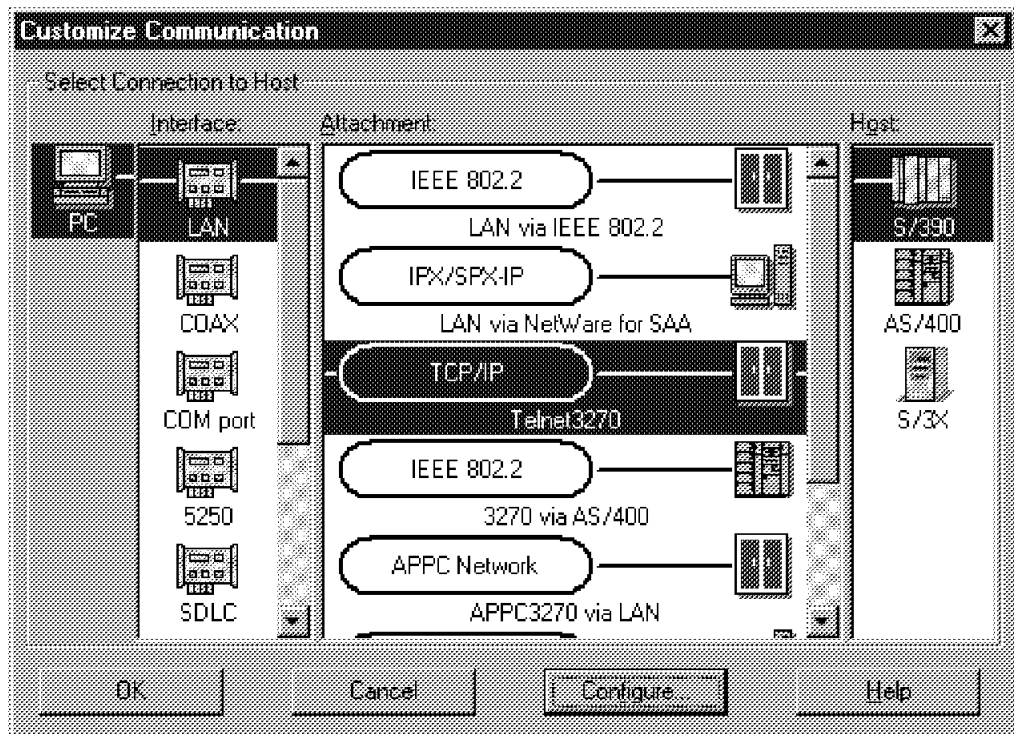


Figure 15. Telnet 3270 Connection Selection

You will select the telnet 3270 option in order to configure the connection to the server (see Figure 16 on page 49). You have the option to enter either a host name or the IP address of the TN3270E server. If you enter a host name, it must be resolved using either a hosts file or a domain name server (DNS).

A backup server can also be configured if required by the client workstation.

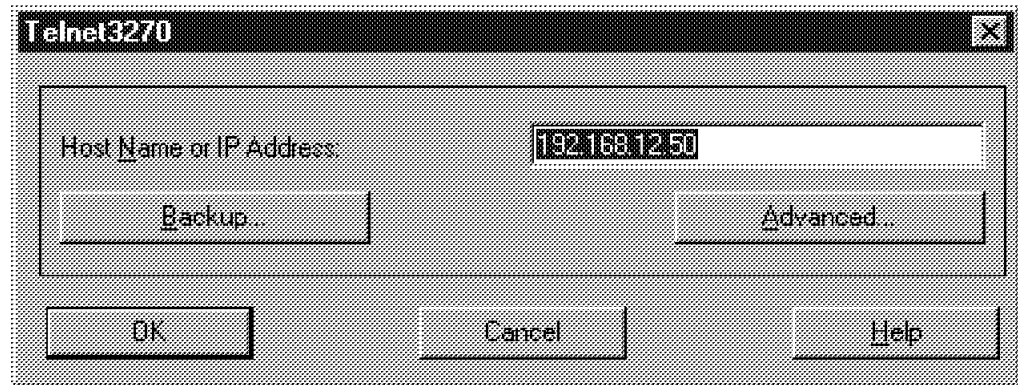


Figure 16. Telnet 3270 Connection Configuration

Optionally, you may want to select the advanced configuration option if you need to change the default port number or enter the name of a specific server LU name you want to use (see Figure 17).

Telnet 3270 is actually a sockets application and therefore it requires a port number. The default well known port number is 23. You will be required to change the port number if there is a conflict with another sockets application using the same port number.

Note

The telnet application default port number is also 23. If you change the port number here, it also has to match the port number in the TN3270E server.

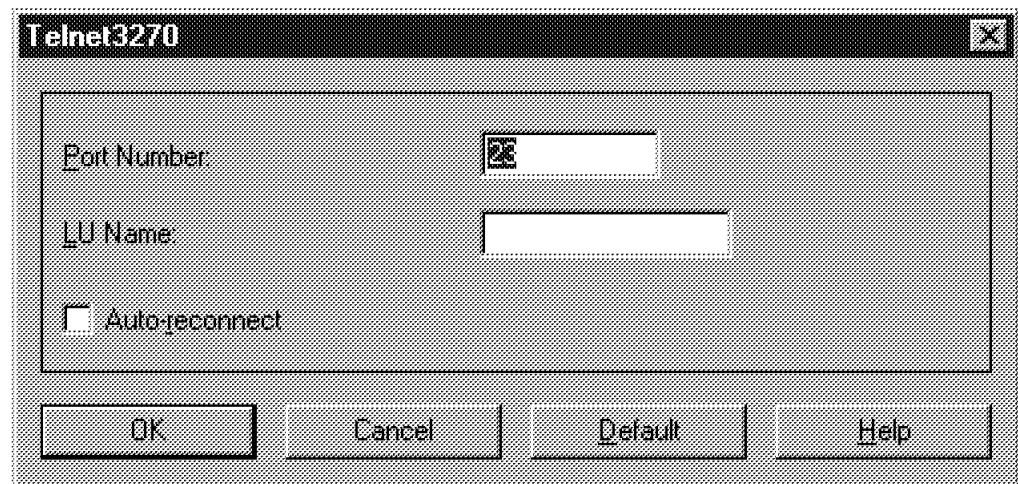


Figure 17. Telnet 3270 - Advanced Configuration

3.7.2 Using 3270 over LAN (IEEE 802.2)

In this section we show you a 3270 emulation session configuration to an SNA gateway using PCOMM in the Windows NT client workstation. SNA traffic over LAN uses the IEEE 802.2 protocol. Therefore, you will select the LAN via IEEE 802.2 configuration option (see Figure 18).

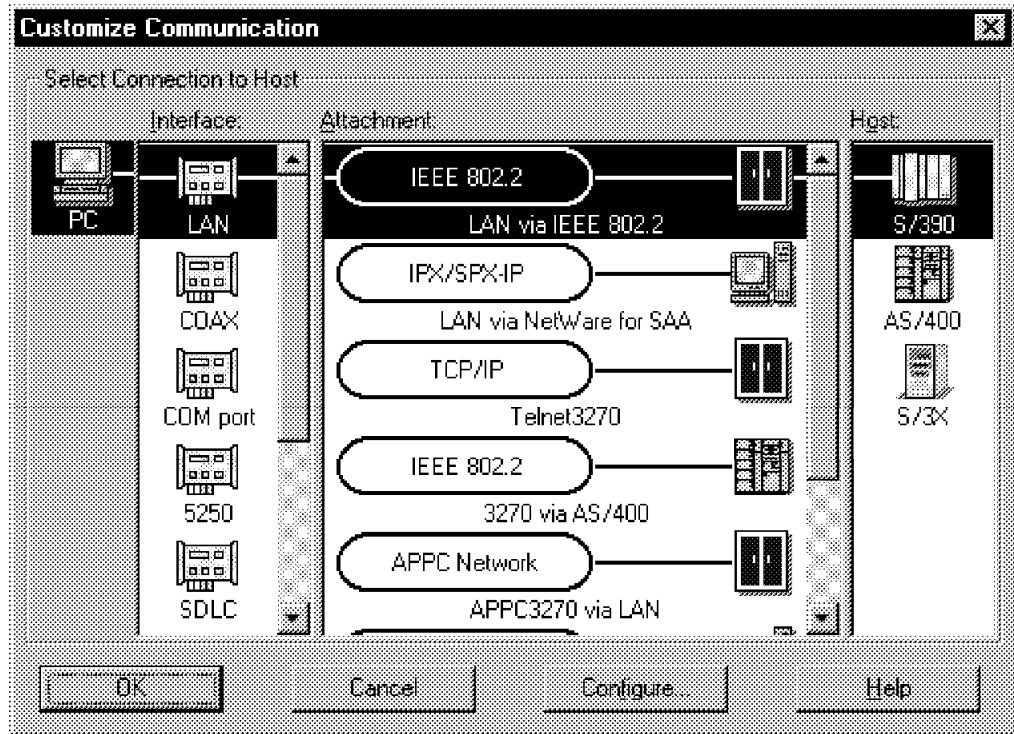


Figure 18. Selecting SNA over LAN for 3270 Emulation

Select Configure... to enter the configuration options as shown in Figure 19 on page 51.

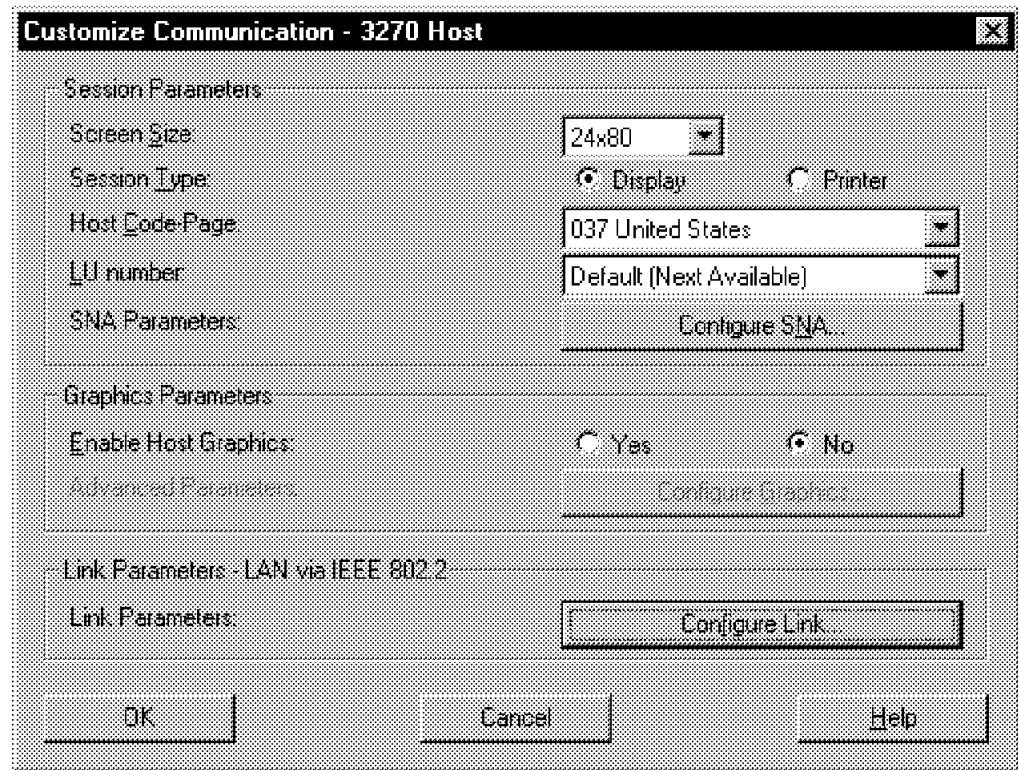


Figure 19. SNA over LAN - Link Parameters

Selecting Configure Link will take you to the Configure Local System panel where you can either configure an existing or new configuration. The panels are very similar to the ones you used to create the configuration of your client workstation for APPC applications (see Figure 20 on page 52).

Configure Local System

SNA Node Configuration File

Existing New Default

D:\Personal Communications\private\wsn101.scg

PC Location Name

Net ID: USIBMRA CP Name: WSNT

Block ID: 05D

Physical Unit ID: 00000

< Back Next > Cancel Help

Figure 20. Local Node Configuration

Next you should configure the LAN device for your 3270 emulation sessions.

Configure LAN device

Device name: LAN0-04

☐ Use first available LAN adapter

Adapter number: 0

Local SAP: 04

< Back Next > Cancel Help

Figure 21. LAN Device Configuration

A logical link or connection will be defined specifying the destination address of the SNA gateway. If you do not know or you are not sure about the target MAC

address, you can always invoke the discovery function as long as your server is enabled to listen for discovery requests. Be aware that discovery is a different IEEE 802.2 application and therefore it uses a different SAP (0xDC).

Configure LAN Connection

Link station name: 0311B

Device name: LAN0_04

Discover network addresses...

Destination address: 400052005198

☒ Tokenring
☐ Ethernet

Remote SAP: 04

Maximum PDU size: 32767

Advanced

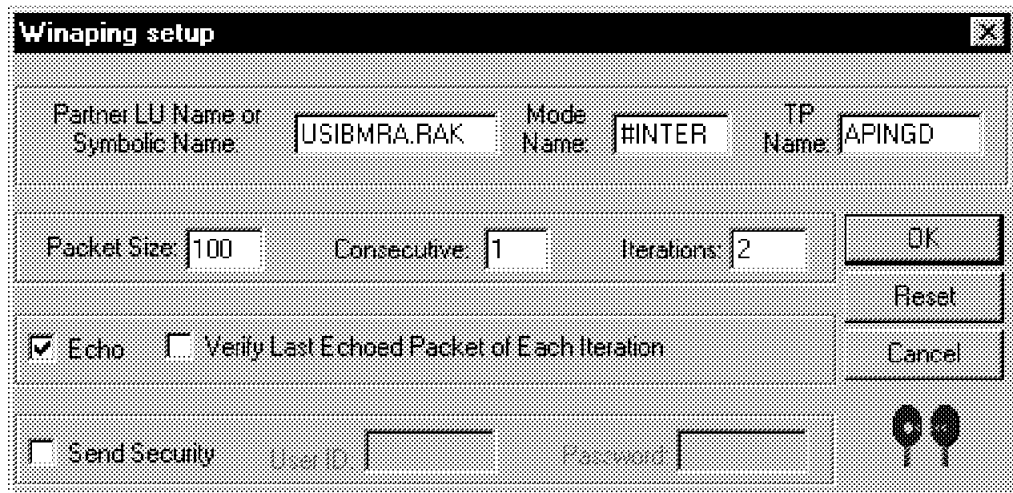
< Back Next > Cancel Help

Figure 22. LAN Connection Configuration

3.8 Monitoring SNA Connections

You will normally use the APING application to make sure you have connectivity with the server. APING is a CPI-C application and uses LU 6.2 protocols.

Select the APING client program and enter the SNA partner LU name of your server. For ease of use you may want to use the default LU names also known as CP-LU. That way you do not have to define local or partner LUs (see Figure 23 on page 54).



Winaping setup

Partner LU Name or Symbolic Name: Mode Name: TP Name:

Packet Size: Consecutive: Iterations:

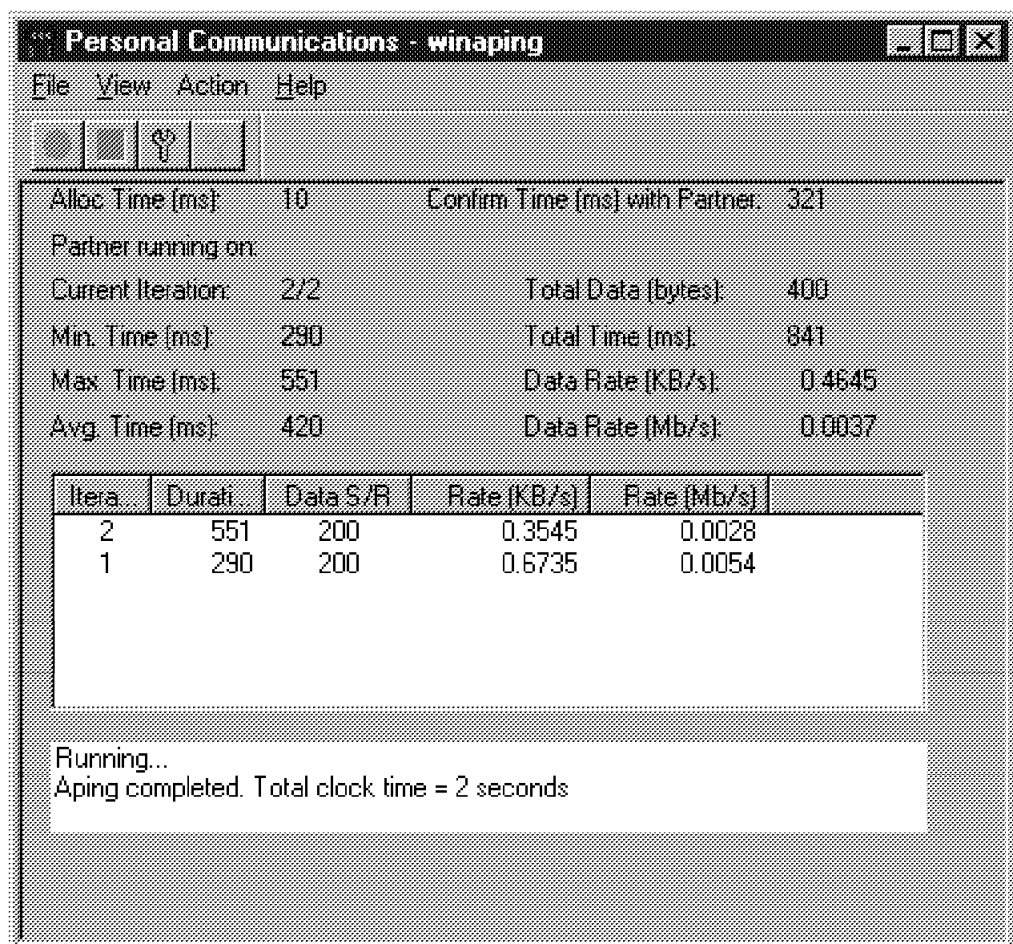
☒ Echo ☐ Verify Last Echoed Packet of Each Iteration

☐ Send Security UserID: Password:

OK, Reset, Cancel buttons and a help icon.

Figure 23. APING Setup

In Figure 24 you obtain the results of your test with detailed information of the operation.



Personal Communications - winaping

File View Action Help

Alloc Time (ms): 10 Confirm Time (ms) with Partner: 321

Partner running on:

Current Iteration: 2/2 Total Data (bytes): 400

Min. Time (ms): 290 Total Time (ms): 841

Max. Time (ms): 551 Data Rate (KB/s): 0.4645

Avg. Time (ms): 420 Data Rate (Mb/s): 0.0037

Itera.	Durati	Data S/R	Rate (KB/s)	Rate (Mb/s)
2	551	200	0.3545	0.0028
1	290	200	0.6735	0.0054

Running...
Aping completed. Total clock time = 2 seconds

Figure 24. Aping Results

You can also use the Node Operations panels to monitor your SNA connections and make sure they are active (see Figure 25 on page 55).

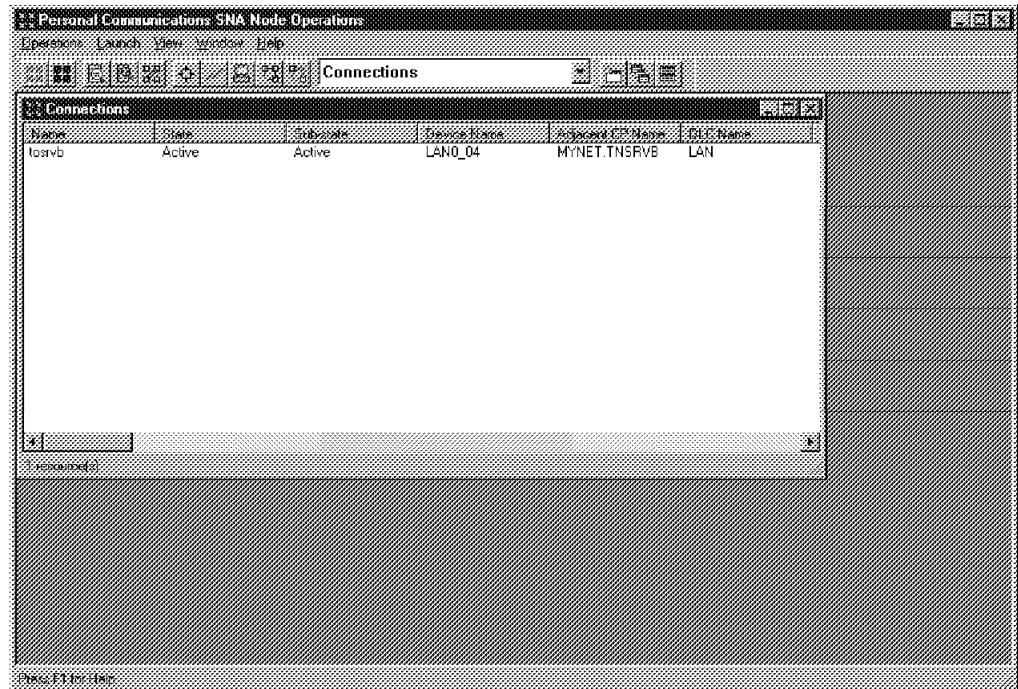


Figure 25. Monitoring SNA Connections

You can also monitor the LU 6.2 sessions as shown in Figure 26. Visually monitor the CP-CP sessions (internal APPN sessions) using mode name CPSVCMG. The internal APPC sessions can be monitored with mode name SNASVCMG and the APING sessions with mode #INTER (default value) or whatever mode was used.

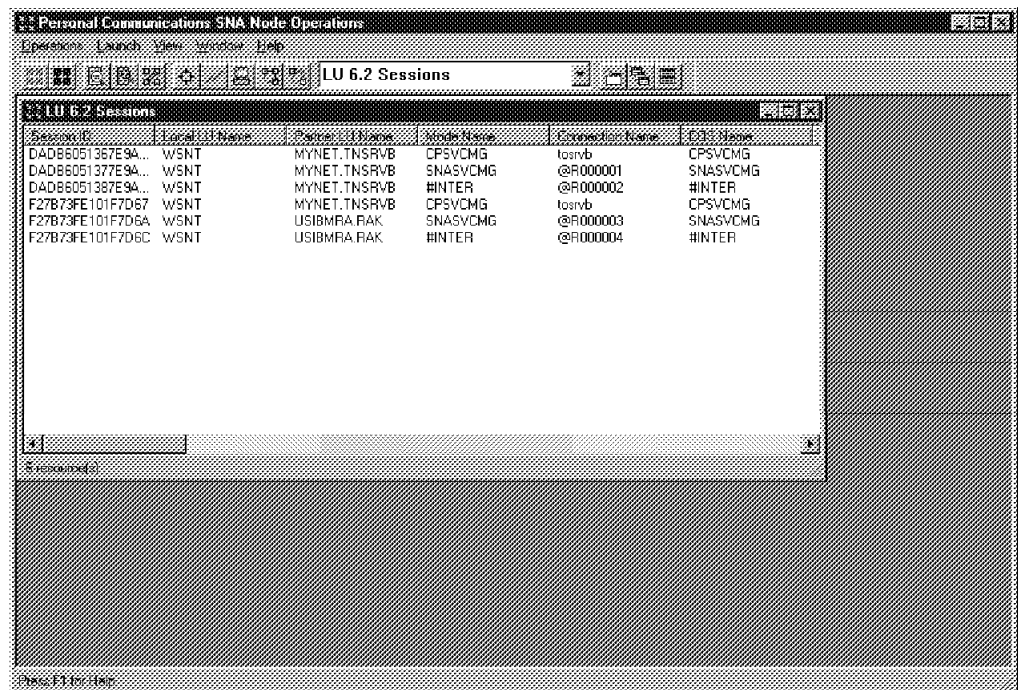


Figure 26. LU 6.2 Sessions Display

3.9 Troubleshooting

If your configuration is not working, there are several options you have to isolate the problem. In this section we describe a few procedures you can invoke in order to find out what the problem is.

3.9.1 Log Viewer

In the log viewer you find all the significant events previously logged. Note that in Figure 27 there is a description, in the lower part of the panel, that gives you more in-depth detail of the problem or event.

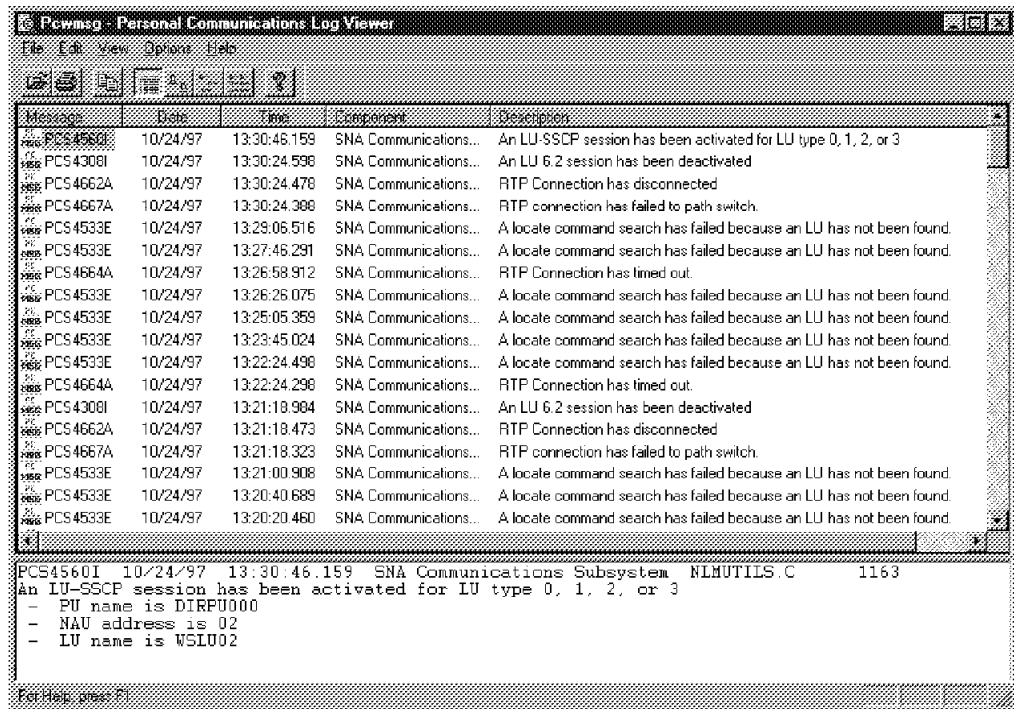


Figure 27. Log Viewer Display

By pressing F1 you can also get additional information as shown in Figure 28 on page 57.

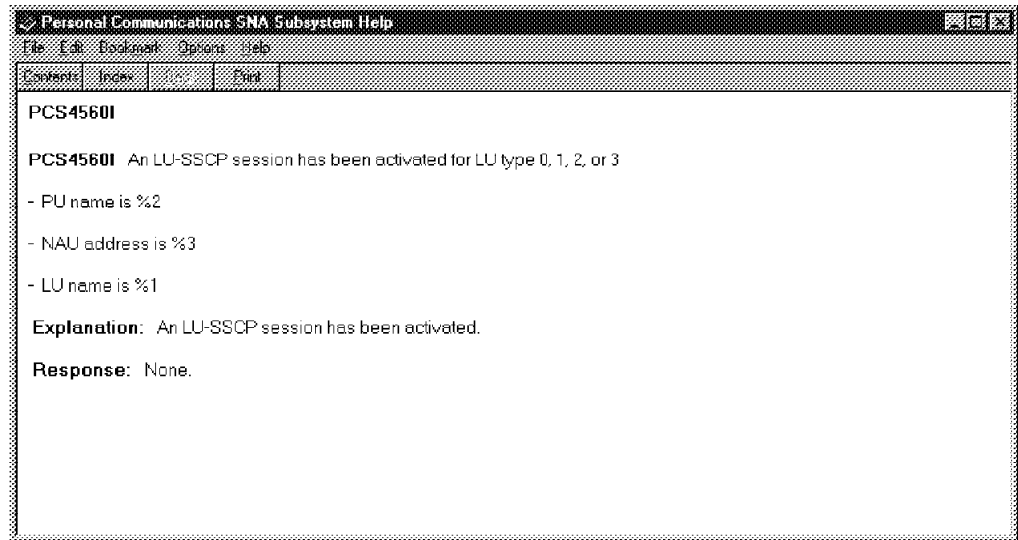


Figure 28. Log Viewer - Help Panel

3.9.2 Traces

The trace facility allows you to set traces for a specific function and component with different options. See Figure 29.

If required you will need to run traces of the scenario that is giving you the problem. If you need to set multiple trace options, use the Apply option to set the options one at a time. After applying the trace options, you will start the traces. Traces should be activated and deactivated using this panel.

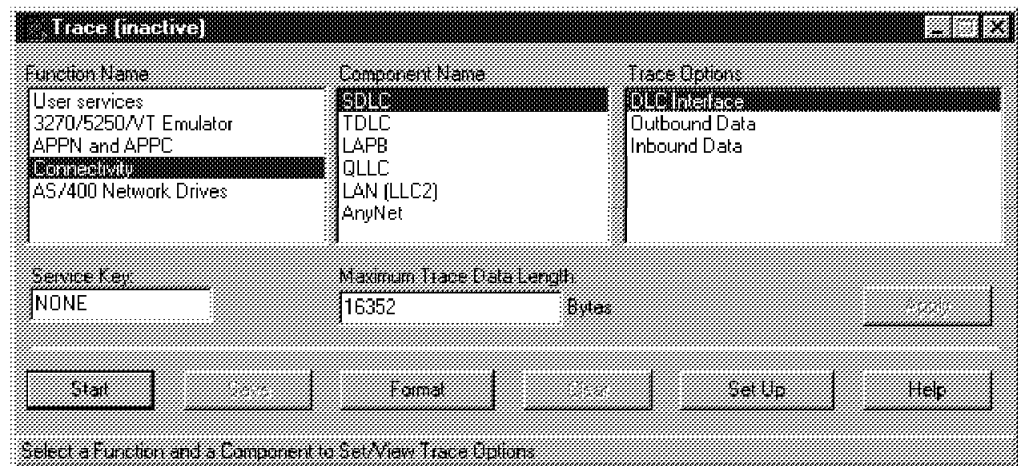


Figure 29. Trace Options

After stopping your traces, you may want to use the Format option to have the traces in a more readable form. You will get an option to enter the target file name or use a default value. See Figure 30 on page 58.

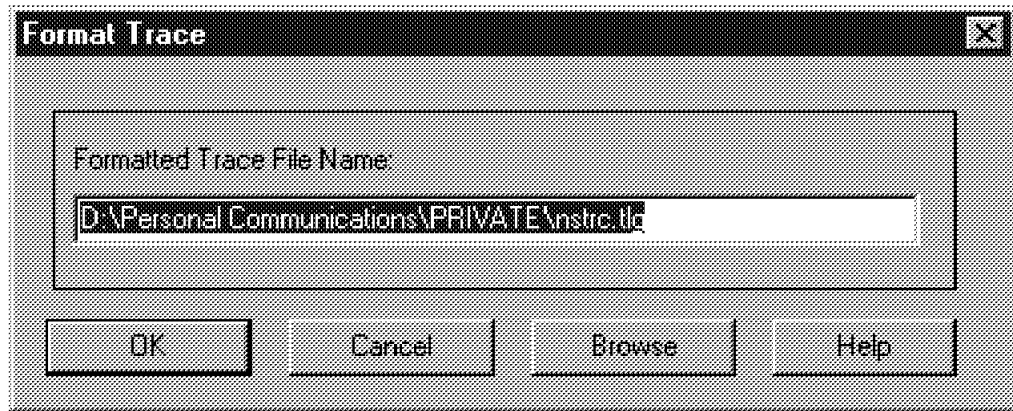


Figure 30. Format Trace Option

3.9.3 GetSense Utility

The GetSense utility program can be used to display the meaning of a specific sense code. When troubleshooting, most of the time you will find the error description in the log viewer and for a better understanding of a sense code, if one is provided, you may want to use this utility. Also, if there is a sense code, as a result of a problem or protocol violation, you will always find the sense code value in the trace. See sample display for sense code 0x0835xxxx in Figure 31.

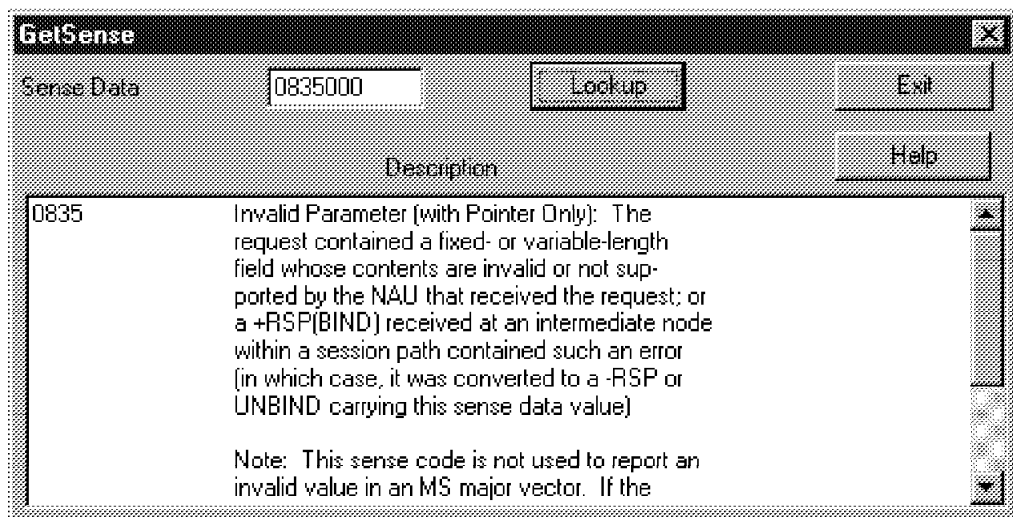


Figure 31. GetSense Utility - Display

Chapter 4. OS/2 Client Workstations

The OS/2 Access Feature provided by eNetwork Communications Server for OS/2 Warp enables applications that are written to the Sockets, APPC, CPI Communications, and LUA APIs to run unchanged over SNA and TCP/IP local and wide area networks. In this chapter we present an overview of the OS/2 Access Feature and describe the installation process.

4.1 OS/2 Access Feature Version 5.0 - Overview

The OS/2 Access Feature also provides applications with local and wide area connectivity support. This is the recommended version for workstations running OS/2 Warp 3.0 or later.

The OS/2 Access Feature provides:

- APPN end node
- HPR: RTP and MLTG support

Note

There is no HPR/ANR support in the OS/2 Access Feature.

- Installation and configuration support
- Local and wide area connectivity support
- Frame relay
- Multiprotocol access node support: SNA over TCP/IP and Sockets over SNA access node

Note

There is no AnyNet Gateway support in the OS/2 Access Feature.

- MPTS
- SNA network support and systems management
- WIN-OS/2 CPI-C support
- 32-bit OS/2 application programming interfaces (APIs)
- APPC and CPI-C client/server APIs
- Dependent LU support for LU 0 to 3 sessions (LU-A)

Note

Although OS/2 Access Feature provides SNA support for dependent LU sessions, it does not include an emulator to provide the presentation services.

For full support of 3270 and 5250 emulation sessions, you should install PCOMM which includes the OS/2 Access Feature.

On a workstation using OS/2, you can install the OS/2 Access Feature. Before installing the OS/2 Access Feature, make sure that you have met the minimum software requirements listed in Prerequisite Software.

The following sections give information on how to install the OS/2 Access Feature using the Communications Server CD-ROM, a code server, diskettes, or a response file.

Consult the OS/2 documentation and application documentation to determine the memory and hard-disk storage required for those parts of your overall installation.

4.2 Prerequisite Software

The minimum software requirements for OS/2 Access Feature are:

- OS/2 Warp Version 3.0 or later
- SystemView support in OS/2 Warp Server and a LAN connection to use the SystemView support in Warp Server for remote installs.
- The version of MPTS that is shipped with Communications Server (or later)
- For multiprotocol transport networking functions, the TCP/IP Sockets access feature of MPTS shipped with Communications Server (or later)

For workstations running OS/2 2.11, you must install the Version 4 access feature. Be sure you follow the instructions for the version you intend to install.

If you want to install the OS/2 Access Feature on a workstation running OS/2 Version 2.11, you must install the Version 4 access feature, provided in the `\CLIENTS\OS2\OS2211AF` subdirectory on the CD-ROM.

4.3 CD-ROM Installation

To install OS/2 Access Feature using the Communications Server CD-ROM, follow these steps:

1. Insert the CD into the CD-ROM drive.
2. From an OS/2 session change directories to the CD-ROM drive.
3. Type `install` or double-click on the **INSTALL EXE** icon from the disk folder.
4. When the Install Program is displayed, click on the Installation icon.
5. When the Installation window is displayed, click on the OS/2 Access Feature icon.
6. Select the **Begin** button.

4.4 Installing Using Response Files

To install OS/2 Access Feature using a response file, follow these steps:

1. Access the product files by doing one of the following things:
 - If you are using a code server, run the commands provided by your administrator to attach to the code server. The commands can include attaching to a LAN drive and changing to the subdirectory where the OS/2 Access Feature files are located (for example, `x:\os2af`).
 - If you are installing from the CD-ROM, change to the CD-ROM drive and either the `\CLIENTS\OS2\WARPAF` (OS/2 3.0 or later) or the `\CLIENTS\OS2\OS2211AF` subdirectory (OS/2 2.11).

- If you are installing from diskettes, insert diskette 1 in the diskette drive. Enter a:. If you are using a different drive, use that drive letter.
2. Enter the following command to start the installation:

```
cmsetup /r response_file /q /l1 error_file /l2 history_file
```

The **/R** response_file specifies the response file name (.RSP).

The **/Q** indicates that the action is unattended; progress indication is not displayed.

The **/L1** error_file specifies the name of the error log file. If you do not specify /L1, then the errors are logged in **CMRINST.LOG**.

The **/L2** history_file specifies the name of the history file. If you do not specify /L2, then the errors are logged in **CM.LOG**.

OS/2 Access Feature is installed and configured based on the response file.
 3. You can also create additional configurations. Configuring Communications Manager and the chapters that follow in the *Quick Beginnings* book on the CD give an overview of the configuration process and the profiles that you might need to define for your configurations. Response files can also be used to define configurations.
 4. Shut down your workstation and restart it.

4.5 Installing Using a Code Server

To install the OS/2 Access Feature using a code server, follow these steps:

1. Run the commands provided by your administrator to attach to the code server. These commands can include attaching to a LAN drive and changing to the subdirectory where the files are located (for example, x:\os2af).
2. Enter cmsetup at the prompt.

You can add parameters to the **CMSETUP** command, but they are generally used only for automated installations or configurations that bypass the installation windows. To view **CMSETUP** parameters, enter cmsetup ? at the OS/2 command prompt.

The Communications Manager logo window and Communications Manager Setup window are displayed.

3. Do one of the following things:
 - If you do not have a configuration or want to modify an existing configuration, select **Setup** to create one and to install the product files.

The interactive configuration interface and configuration profiles are described in Configuring Communications Manager and the chapters that follow in the *Quick Beginnings* book on the CD. Online help is available for each window by selecting the Help push button.
 - If you are upgrading a configuration from Communications Manager/2 or have an existing configuration, select **Installation** to install the necessary product files. Configurations from Communications Manager/2 will be upgraded automatically.

4.6 Installing Using Diskettes

To install OS/2 Access Feature using diskettes, follow these steps:

1. Insert the first diskette in the diskette drive.

For more information about Creating Diskettes see 3.4.1, "Creating Diskettes" on page 34.

2. Open an OS/2 window. If you are installing from diskettes in drive A:, type `a:\cmsetup` and press Enter. If you are installing from another drive, replace the A: with that drive.

You can add parameters to the **CMSETUP** command, but they are generally used only for automated installations or configurations that bypass the installation windows. To view **CMSETUP** parameters, enter `cmsetup ?` at the OS/2 command prompt.

The Communications Manager logo window and Communications Manager Setup window are displayed.

3. Do one of the following things:

- If you do not have a configuration or want to modify an existing configuration, select **Setup** to create one and to install the product files.

The interactive configuration interface and configuration profiles are described in *Configuring Communications Manager* and the chapters that follow in the *Quick Beginnings* book on the CD. Online help is available for each window by selecting the **Help** push button.

- If you are upgrading a configuration from Communications Manager/2 or have an existing configuration, select **Installation** to install the necessary product files. Configurations from Communications Manager/2 will be upgraded automatically.

4.6.1 Creating Diskettes

Diskette images for the access features are stored in platform-specific subdirectories on the CD-ROM.

Be sure to make copies of all products needed to install a particular access feature. The software prerequisites are listed in *Introduction to Installing Access Features and Web Features* in the CD.

- OS/2 Access Feature requires that MPTS (shipped on the CD-ROM) be installed and configured.
- APPC Networking Services for Windows requires the IBM Local Area Network Support program (shipped on the CD-ROM).
- AnyNet APPC over TCP/IP for Windows requires APPC Networking Services and one of the TCP/IP products listed under AnyNet APPC over TCP/IP for Windows.

To create diskettes to install one or more of the access features, follow these steps:

1. Open an OS/2 window. If you are creating diskettes from the CD-ROM drive in drive E:, change to the directory that contains the diskette images that you want, for example:

`e:\diskimgs\clients\xxxxxxx`

where xxxxxxx is: OS2\WARPAF, OS2\OS2211AF, WINDOWS\ANYNET, WINDOWS\NETSUPP, or WINDOWS\NSW.

If you are creating diskettes from another drive, replace the E: with that drive.

2. Determine the number of diskettes needed by entering dir at the command prompt.
3. Use the LOADDSKF command to load a copy of the diskette images onto diskettes. For example, to load the first diskette for APPC Networking Services, enter:

```
e:\diskimgs loadskf nswinst1.dsk a: /f
```

The /F indicates that the diskette will be formatted to match the input image.

LOADDSKF is located in the \DISKIMGS subdirectory. To get help on this command, enter loadskf at the command prompt.

4.7 OS/2 Access Feature - Sample Installation

In this section we describe a sample OS/2 Access Feature installation. You can install the OS/2 Access Feature from the eNetwork Communications Server for OS/2 Warp CD or from the PCOMM CD. In either case, you get support for APPC, CPI-C and LU-A applications but you need to be aware that eNetwork Communications Server for OS/2 Warp does not provide the 3270 and 5250 emulators for client workstations.

4.7.1 MPTS Requirements

MPTS is distributed in the eNetwork Communications Server for OS/2 Warp CD-ROM. It provides the support for LAN and some WAN adapters (for example, IBM WAC adapters). Also, MPTS provides the IEEE 802.2 protocol to support SNA over LAN traffic and frame relay. Other WAN protocols, such as SDLC and X.25 are provided by eNetwork Communications Server for OS/2 Warp.

If your MPTS level is not at release 5.12 the installation will prompt you to install it before it can continue. Failure to do so will bring you errors and unexpected behaviors. Normally this will happen in most of the installations. After the MPTS update you should return to the CMSETUP and the installation main screen in order to install the OS/2 Access Feature.

4.7.2 OS/2 Access Feature Installation

In our sample installation we show you how to install the OS/2 Access Feature from the eNetwork Communications Server for OS/2 Warp CD-ROM. The procedure is very similar if you install from the PCOMM CD-ROM.

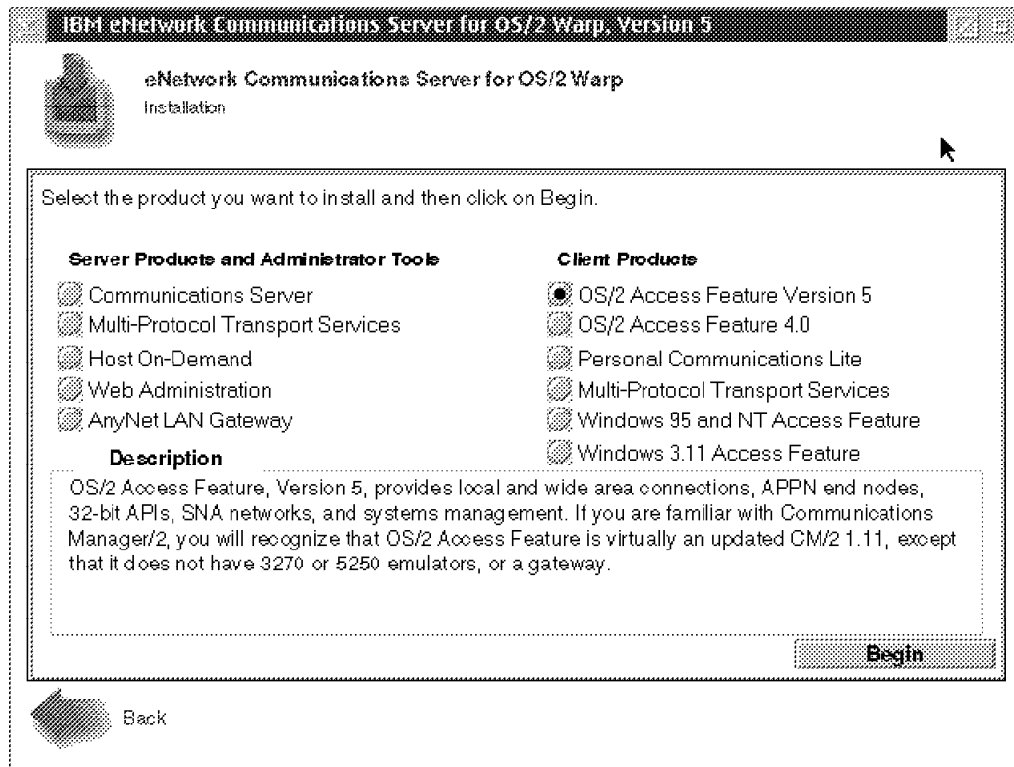


Figure 32. OS/2 Access Feature Installation



Figure 33. OS/2 Access Feature Installation and Setup

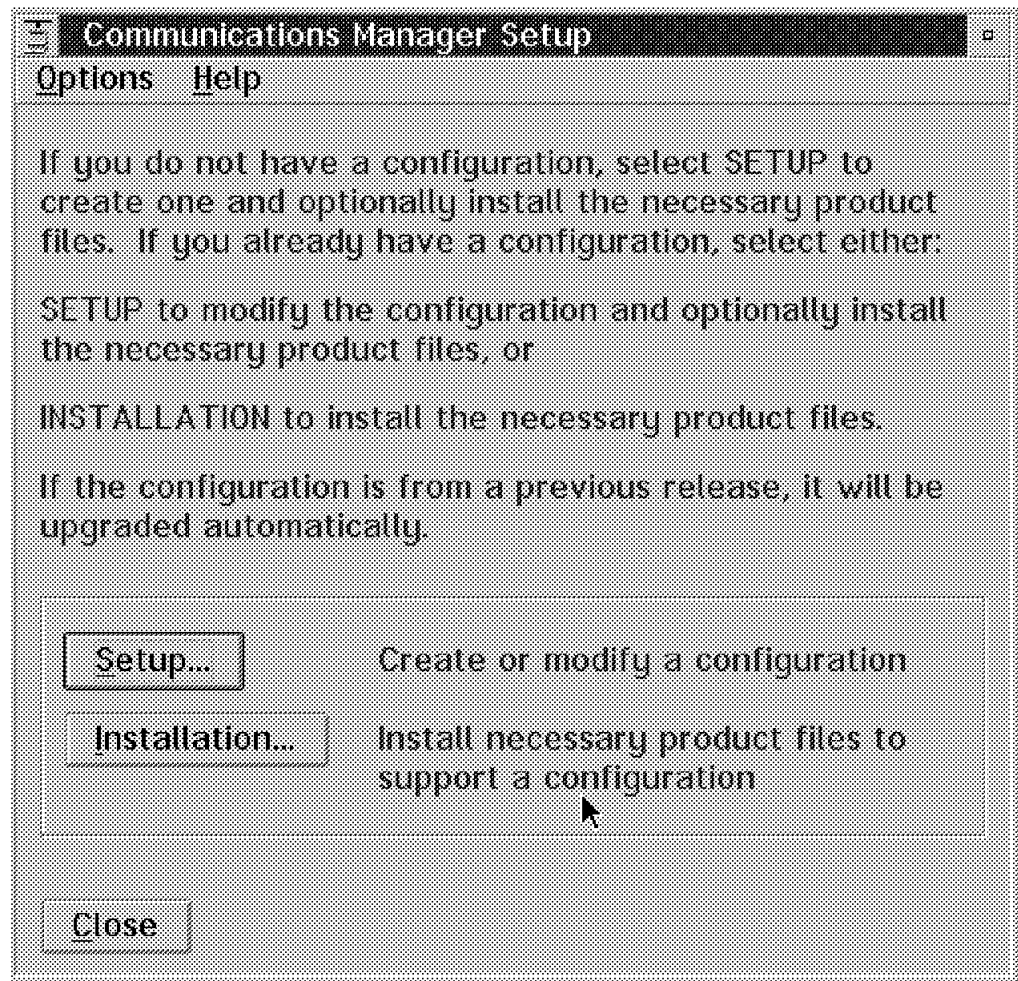


Figure 34. Setup

Select whether you want to set up (configure) or install Access Feature.

Select Setup if you want to:

- Create a new configuration
- Modify an existing configuration
- Upgrade and modify a configuration that was created with a previous version.

In Setup, you make selections and provide information about your workstation and your network that form the basis of your configuration. Access Feature uses this information to manage your network communications.

When you finish creating or modifying a configuration, you get the option to update the libraries with the modules required to support your new configuration. You do not need to return to this window for installation of the files needed to support your new configuration.

Select Installation if you have an existing configuration that you want to use on this workstation. This can be a configuration that was created on another workstation, or it can be a configuration created in a previous release of the product. Installation is the process of copying Access Feature product files,

updating the CONFIG.SYS file, and updating desktop folders with the functions you need to make a particular configuration work on your workstation.

4.8 OS/2 Access Feature - Sample Configuration

In this section we show you a basic OS/2 Access Feature configuration. An OS/2 workstation is configured as an end node (EN) and it connects to a network node (NN) or to a branch extender node (BrNN). This configuration will allow you to run client/server APPC or CPI-C applications as well as any of the LU-A applications including PCOMM emulation sessions.

To open an existing configuration, you can type its name in Configuration, or you can select it from the Configurations list. When you select a configuration from the list, the name appears in this field.

We now create a configuration to connect our end node workstation to a network node server which will also be our gateway for 3270 sessions.

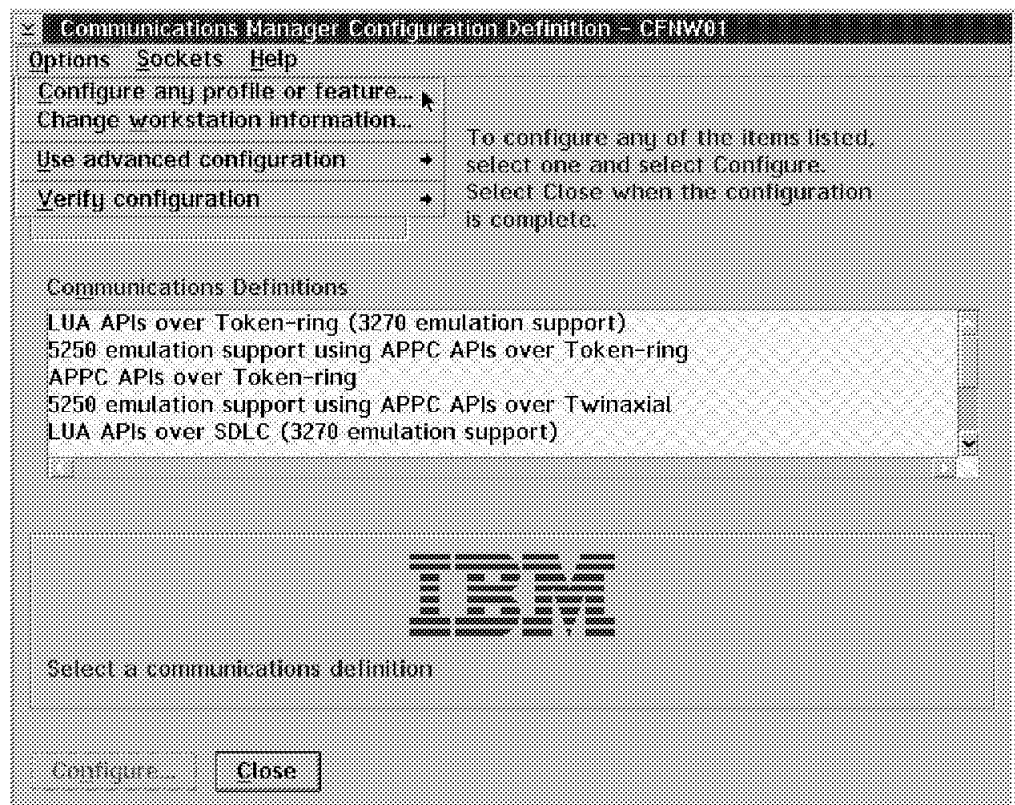


Figure 35. Configuration Selection

Communications Definitions lists the following feature or application and connection type combinations that you can configure quickly using the supplied default values:

- LUA APIs over Token-ring (3270 emulation support)
- 5250 emulation support using APPC APIs over Token-ring
- APPC APIs over Token-ring
- 5250 emulation support using APPC APIs over Twinaxial

- LUA APIs over SDLC (3270 emulation support)
- 5250 emulation support using APPC APIs over SDLC
- LUA APIs over Ethernet Network (3270 emulation support)
- 5250 emulation support using APPC APIs over Ethernet Network
- APPC APIs over Ethernet Network

The following list describes the available connection types in more detail.

Note: Some of the workstation connection types described in this list may not appear in the Configuration Definition window on your workstation.

You will normally find the most common scenarios in the communications definition section of the main configuration panel. If you do not find your scenario, you will have to select the **Configure any profile or feature** for advanced configuration. The following DLCs are available in the communications definitions:

- **Token-ring (or other LAN types)** applies to a workstation connected to a LAN using an adapter based on the network device interface specification (NDIS) for LAN protocols. These networks include token-ring, BusMaster, 3174 Peer Attachment (for LANs connected by coaxial cable), fiber distributed data interface (FDDI), and ISDN LANS.

Note: This choice does not include Ethernet networks or IBM PC Networks (or similar). See the separate choices in this list for these workstation connection types.

- **Twinaxial for AS/400** applies to a workstation connected to a network controller by a twinaxial cable.
- **SDLC** applies to a workstation connected to a network linking a variety of devices that use synchronous data link control (SDLC) facilities. SDLC uses leased lines to establish manual-dial connections through a multi-protocol adapter (MPA) card. Most high-speed mainframe computer networks are of this type.

This connection type supports permanent and DTR/manual connections over synchronous adapters such as the IBM WAC, MPA, and ARTIC adapters. This connection type also supports permanent connections over asynchronous ports or adapters such as serial (COM) ports or ARTIC adapters in asynchronous mode.

- **Ethernet (ETHERAND) network** applies to a workstation connected to a LAN that uses the Ethernet Digital Intel Xerox (DIX) Version 2.0 protocol.

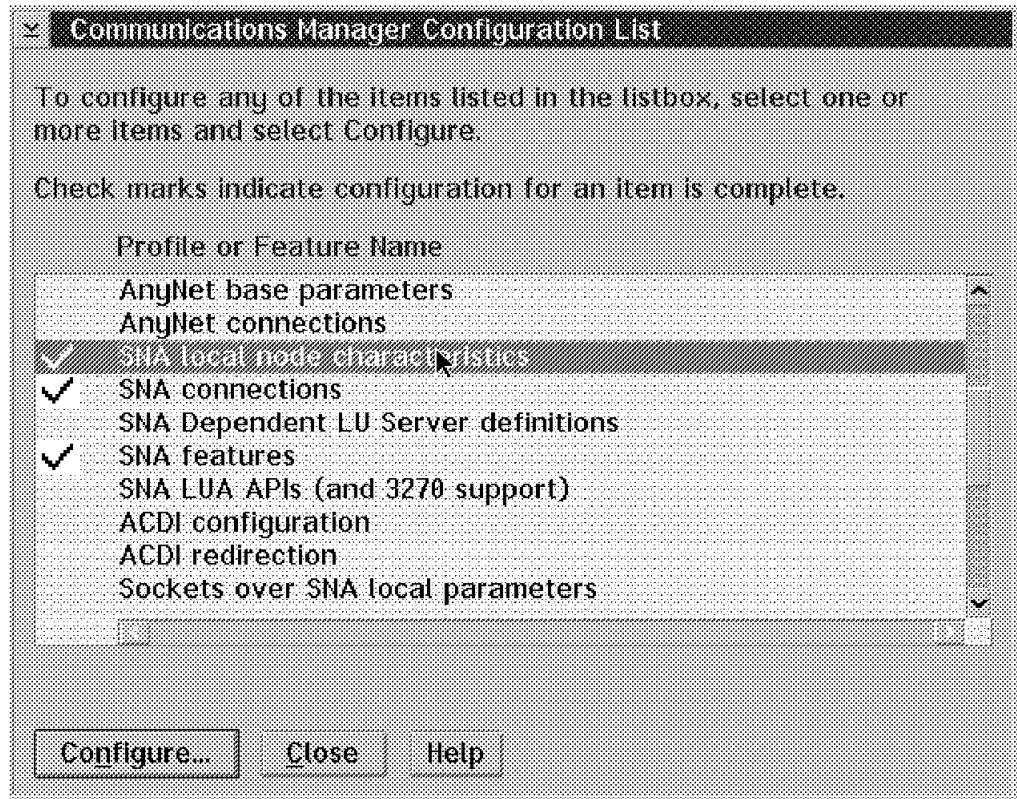


Figure 36. Configure Any Profile or List - Configuration List

Profile or Feature Name lists all of the profiles and features you can configure using Communications Manager. The ones with check marks next to them are already defined in this configuration.

Select the profile or feature you want to configure or change. You can select several choices at once. Then select Configure...

When you select Configure..., it consecutively displays the windows required for each profile or feature you selected.

For descriptions of the information you need to configure each of the profiles or features, select from the following list:

Note: Some of the profiles and features described may not appear in the configuration list on your workstation.

- DLC (data link control) profiles
- SNA (Systems Network Architecture) profiles
- ACDI (asynchronous communications device interface) profiles
- SNA gateway profiles
- X.25 profiles
- SNA Phone Connect profiles
- AnyNet profiles
- ARTIC adapter profile

4.8.1 Local Node Characteristics

This profile is mandatory and you always need to configure the local node characteristics for your workstation.

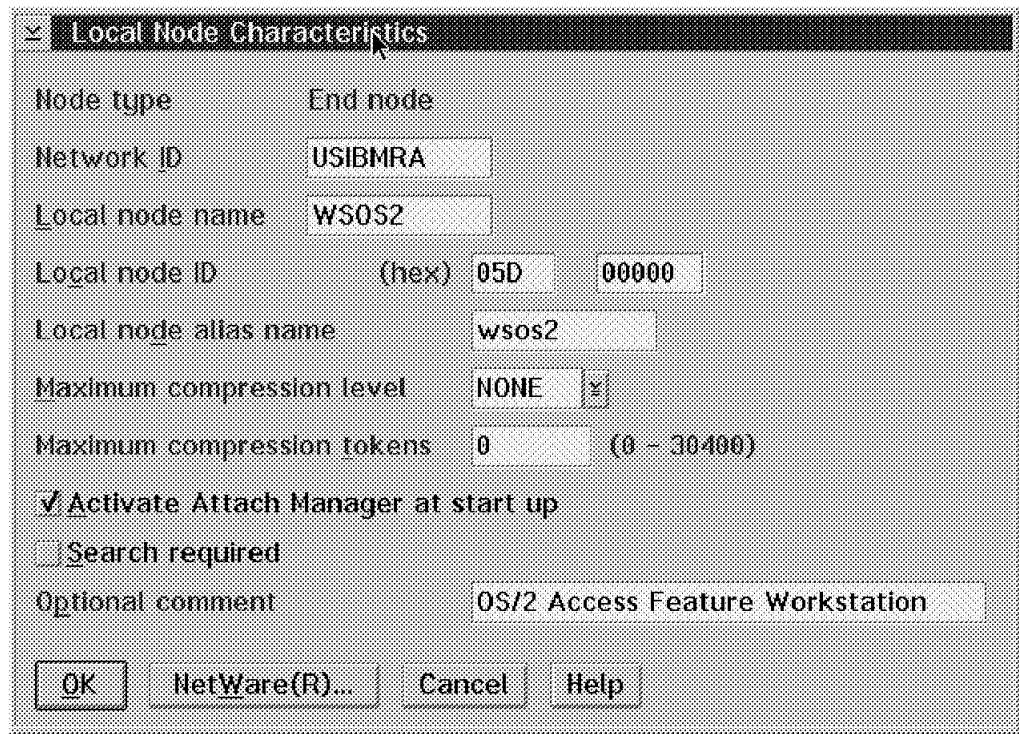


Figure 37. Local Node Characteristics

In our configuration we enter the following values:

Network ID defines the name of your Advanced Peer-to-Peer Networking (APPN) network. You will normally enter the name of the network ID in which your local LU resides.

Note

In SNA networks only network nodes (NNs) are required to have exactly the same network ID.

Nodes within a network use the network ID as part of their node name. Together, the network ID and the node name (fully qualified control point (CP) name) uniquely identify a node within an interconnected network environment. The network ID and the node name also uniquely identify a node for receiving error logs and network management alerts.

Local node name is the name that other nodes in your network use to address your node. Your local node name is also called your control point (CP) name.

Note: OS/2 Access Feature uses the local node name, or control point name, as the default local LU for your workstation, unless you specify another default in the NDF configuration file.

Your local node name must be unique.

Enter the local node name for your workstation. When assigning a local node name, consider the following rule: when you change the local node name, the name in local node alias stays the same unless you change it.

Note: If you are configuring management support for NetWare, the local node name is used as the user ID for creating a connection to a NetWare server. The user ID and its related password must be configured on the NetWare server.

Local node alias name defines an optional alternative name, or alias name, by which local programs can refer to your node, or control point LU. Alias names are generally used for convenience or easy recognition within a network.

You can type both uppercase and lowercase letters in the Local node alias name field. However, the system saves the name exactly as you enter it. For example, if you enter a name using a combination of uppercase and lowercase letters, the name is saved in the same manner.

Note: If this field is left blank, local node name is used as the local node alias.

4.8.2 Adapter Configuration - Devices

For Token-Ring or Other LAN Types DLC Adapter Parameters (Figure 38) you need to select the adapter you will use.

Token Ring or Other LAN Types DLC Adapter Parameters

Adapter (0 - 15)

☐ Free unused links

☐ Branch extender support

Maximum I-field size
 (265 - 16393)

Local SAP (hex)
 (04 - 9C)

Additional parameters

- HPR parameters
- Link initialization parameters
- Link station protocol parameters
- Network management parameters
- Resource parameters

Effective capacity (bits per second)

Connection network parameters (optional)

Name . ☐ Limited resource

Figure 38. Adapter Parameters

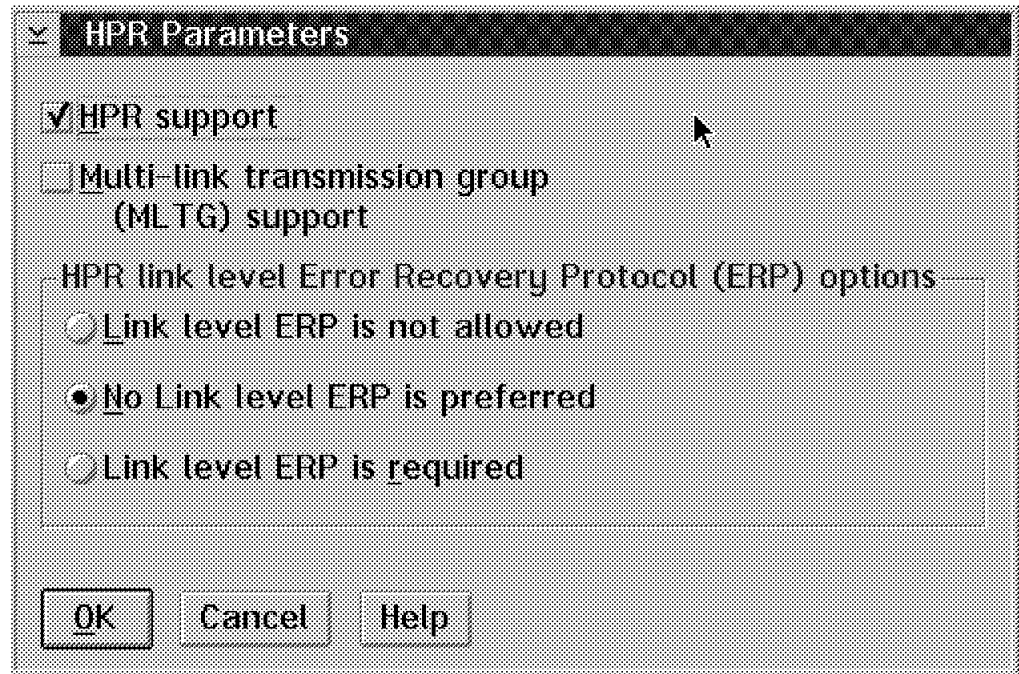


Figure 39. HPR Parameters

It is important that you select the HPR parameters and click on the HPR Support checkbox (Figure 39).

4.8.3 Logical Link Configuration - Connections

We are connecting to a network node, so we select this option. By selecting Create you are presented with the panel shown in Figure 41 on page 73 where you will select the type of adapter. If you select token ring and then Continue you obtain the screen shown in Figure 42 on page 73.

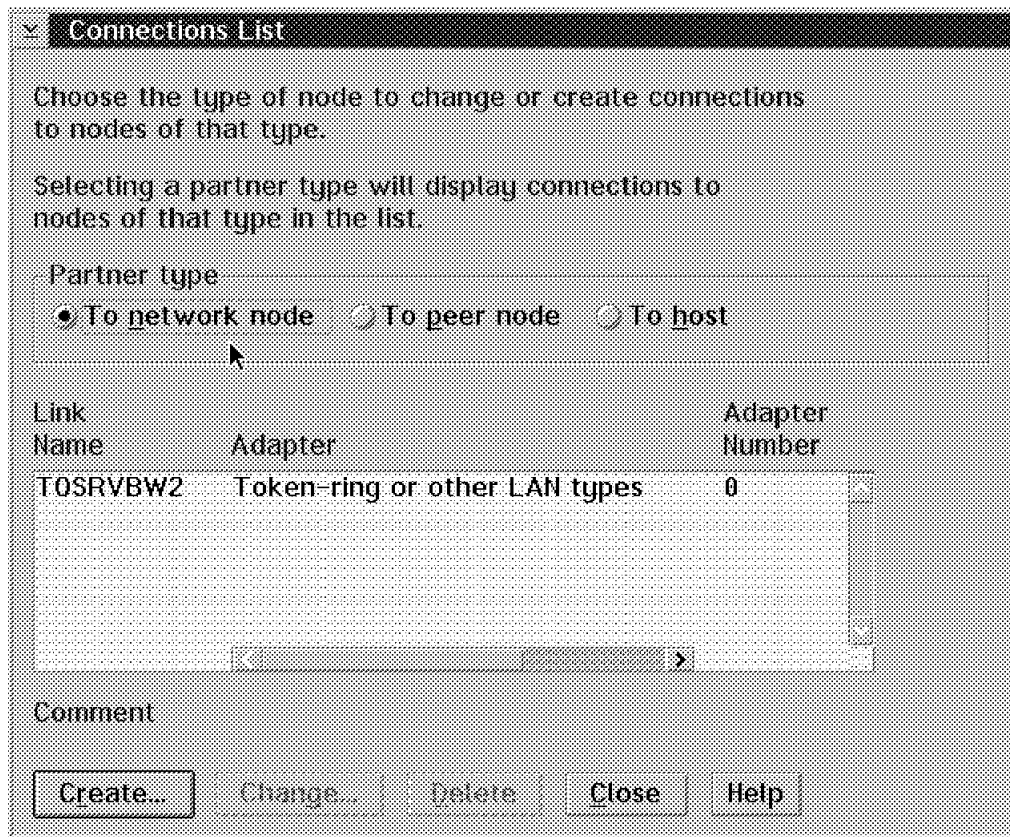


Figure 40. Connections List

If you plan to have dependent LU sessions, such as 3270 emulation sessions, and you do not have DLUR in your workstation, your connection must have the option to activate the SSCP-PU session. This option is always enabled when you select a **To host** connection. However if you configure an APPN connection, you need to do this manually in the configuration of the logical link.

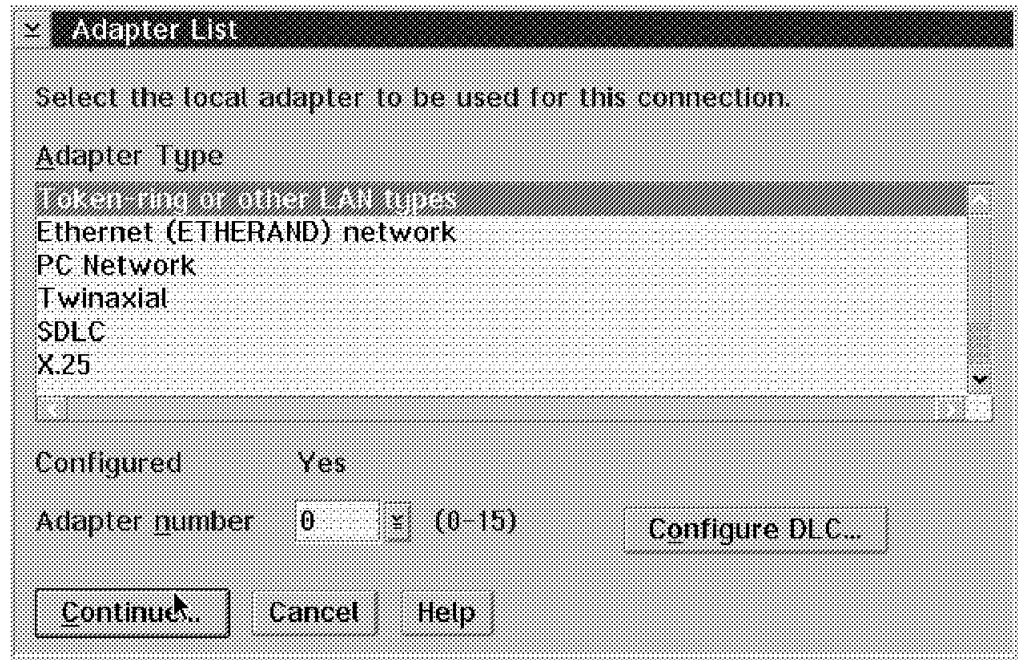


Figure 41. Adapter Selection

Adapter List contains a list of adapter numbers that have been configured. You can either select a number, or type an adapter number in the field. When you select an adapter, the current configuration for that adapter appears in the window. You can then change the configuration information.

The LAN adapter on your workstation has an adapter number assigned to it when you configure MPTS. The adapter number you select or type should match this number.

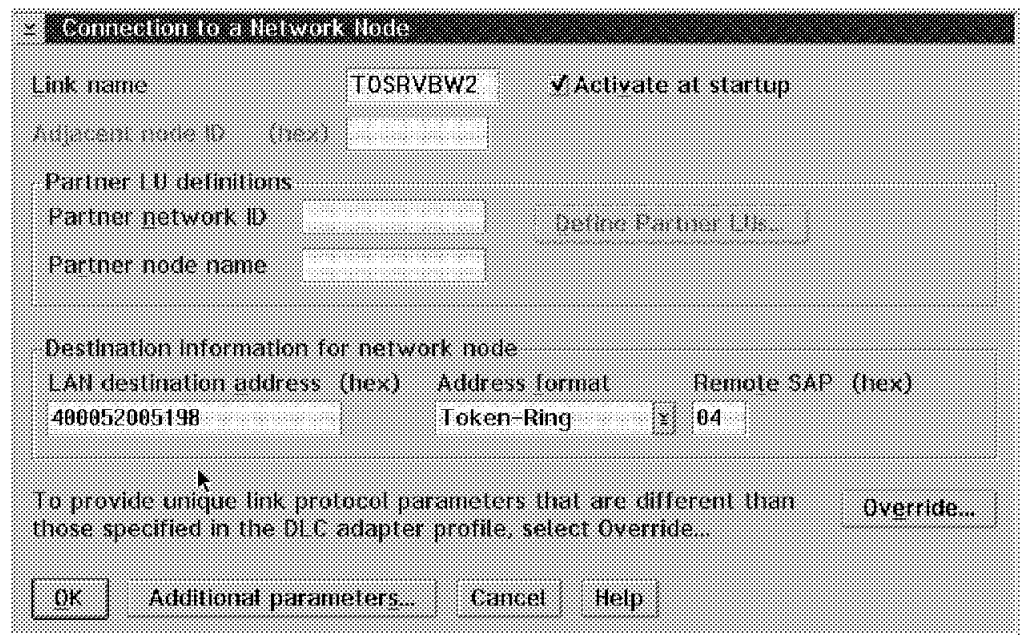


Figure 42. Connection to a Network Node - Configuration Panel

LAN destination address (hex) contains the 12-character hexadecimal address of the adapter on the node to which you are defining this connection. When you type the address, do not enclose the value in the hexadecimal symbols. For example, enter 01AB, not X'01AB'.

Type the address in token-ring format. If the partner you are communicating with uses Ethernet format, select Ethernet from the Address format list.

Depending on your network, the LAN destination address is one of the following:

- The node address or the media access control (MAC) address defined in the configuration for the controller's network adapter on an IBM Token-Ring network or an IBM PC network
- A permanent address that is encoded in the controller's network adapter by the manufacturer
- An address assigned within your network to the controller's network adapter

Your LAN administrator can tell you the LAN destination address to use.

If you enter a value in the Adjacent node ID field, you cannot enter a value in the LAN destination address field. To complete this window, you need to enter either an adjacent node ID or a LAN destination address.

We will connect using a token-ring card, in this case the MAC address of our server is 400052085198.

Optionally select **Override** in this screen (Figure 42 on page 73) to verify that the HPR support has been selected in the correct way.

Then in the Override panel (Figure 43 on page 75) verify the HPR support you want to include in your configuration.

Link Station Protocol Parameters Override

To override the value specified in the DLC adapter profile for any of the parameters below, select the checkbox, and then enter a value for the parameter.

☐ Maximum I-field size (265-16393)

☐ Effective capacity (bits per second)

☐ Branch extender support ☐

☒ HPR support ☐

HPR link level Error Recovery Protocol (ERP) options

☐ Link level ERP is not allowed

☐ No link level ERP is preferred

☐ Link level ERP is required

☒ Use adapter definition

Multi-link transmission group (MLTG) number

☒ Default TG number

☐ Explicit TG number (1 - 20)

Figure 43. Link Station Protocol Parameters - Override

After saving and closing your configuration, CMSETUP will verify the values you entered for proper syntax and consistency.

If new function is added to your configuration, CMSETUP will prompt you to insert the CD-ROM in order to install the new modules in your machine.

After the installation process, you will be asked about the way you want the folders of the access feature to be created, and the option to update the config.sys.

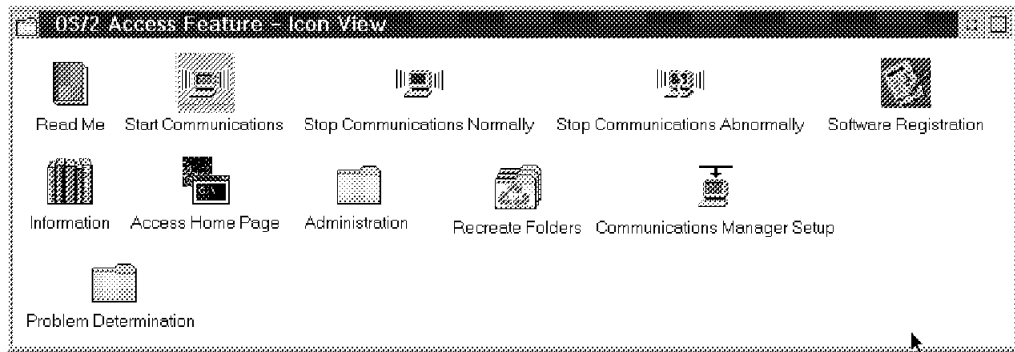


Figure 44. OS/2 Access Feature Folder

Once installed, opening the Access Feature folder on your Desktop will get you all the icons shown in Figure 44

4.8.4 Independent LU Configuration for APPC

Independent LUs are used in different ways to establish LU 6.2 sessions between an OS/2 workstation and a communications server. For example, in OS/2 Access Feature, independent LUs are used by:

- APPN internal control sessions. Also known as CP-CP sessions.
- APPC internal control sessions. CNOS commands flow over these sessions.
- Application traffic using the APPC, CPI-C or CPI-C for Java. We normally refer to these applications as client/server applications.
- Sockets over SNA transport sessions.

4.8.4.1 Independent Local LUs

In this sample configuration we use the default independent LU (also known as the CP-LU) to run our client/server applications. Therefore, there is no need to define any extra independent local LUs.

4.8.4.2 Independent Partner LUs

In this sample configuration our APPC applications do not use alias names and the connection from the OS/2 workstation (EN) to the communications server machine (NN) is via APPN.

The APPN mission is to locate partner LUs and for this reason we do not need to configure partner LU locations. Therefore, there is no need to define any partner independent LUs.

4.8.5 LU-A Profiles

You will need to configure LU-A profiles when you need support for dependent LU sessions (LU 0 to 3) in your workstation. You can create specific dependent LUs or you can do multiple LU definitions if required.

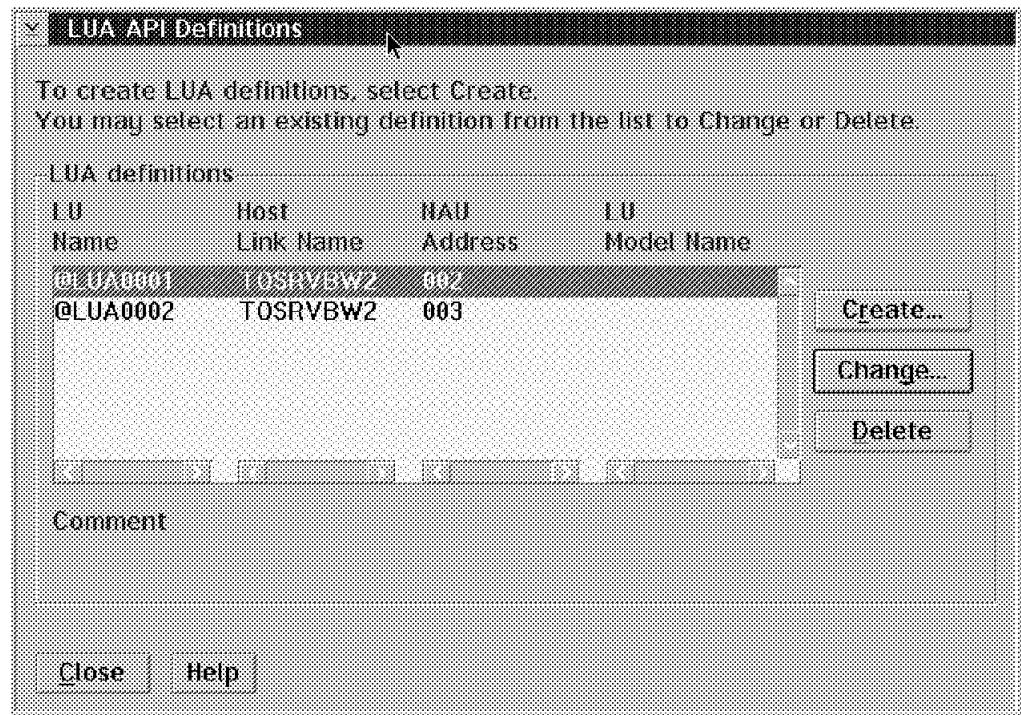


Figure 45. LU-A Profile Selection

This list contains the LUA API definitions that are defined. You can create a definition by selecting Create. You can change or delete definitions by selecting the definition and then selecting Change or Delete.

- **LU Name** defines the name of the Systems Network Architecture (SNA) logical unit application (LUA).
- **Host Link Name** is the name of a logical link definition that connects a node that supports any systems services control point (SSCP) driven system.
- **NAU Address** is the network addressable unit (NAU) address that is the source or destination of information transmitted by the path control network. This address must be unique per host link.
- **LU Model Name** is optional and is used by VTAM to dynamically define the dependent LU at the host.
- **Comment** can help you remember the contents and purpose of this definition.

LUA API Parameters

LU name: @LUA0001

Host link name: TOSRVBW2

NAU address: 002 (001 - 254)

Optional LU
model name:

Optional comment:

Figure 46. LU 0 to 3 Configuration Panel (LU-A)

4.8.6 Installing Personal Communications Version 4.2

As we have stated previously the Personal Communications Entry Level version included in eNetwork Communications Server for OS/2 Warp Version 5.0 provides basic support for emulation sessions. In addition, the distributed PCOMM Entry Level emulator is not available for client workstations running the OS/2 Access Feature only. Therefore, for full emulation session support you should install PCOMM.

In this section we show how to configure 3270 emulation sessions using PCOMM for presentation services support and LU-A from OS/2 Access Feature for SNA stack support.

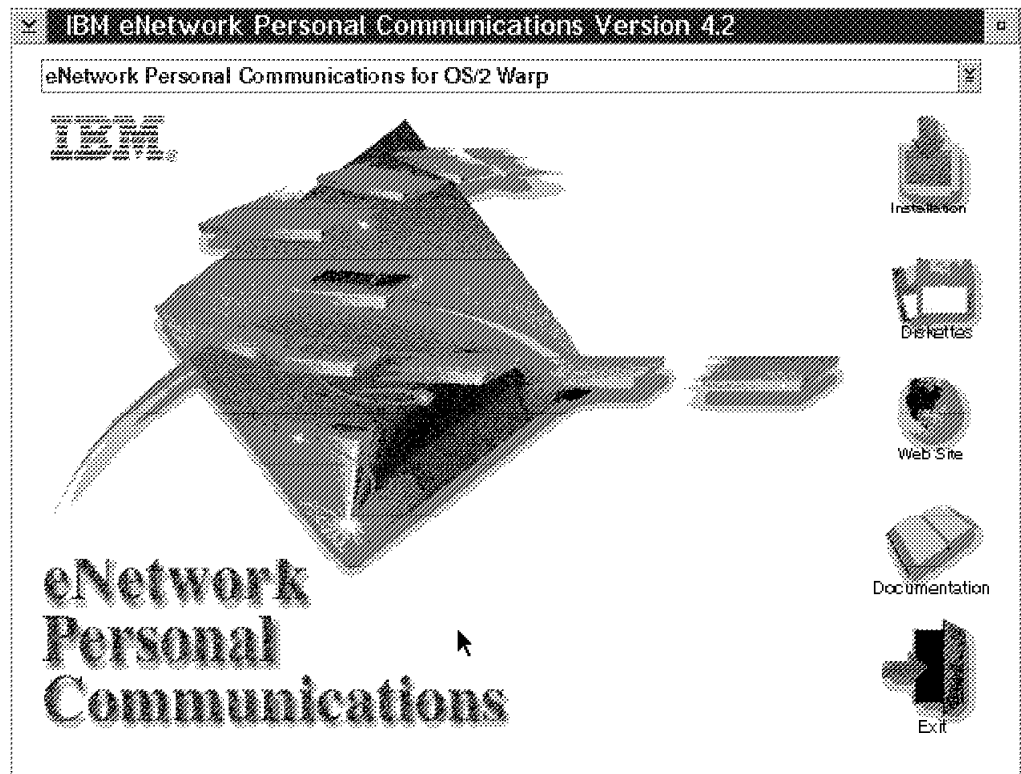


Figure 47. Personal Communications - Installation

4.8.6.1 PCOMM via LU-A Configuration

If the SNA stack support is provided by LU-A in OS/2 Access Feature, you will select the LU-A names previously configured using OS/2 Access Feature CMSETUP.

The LUs can be associated with a subarea PU or DLUR PU when using APPN networks.

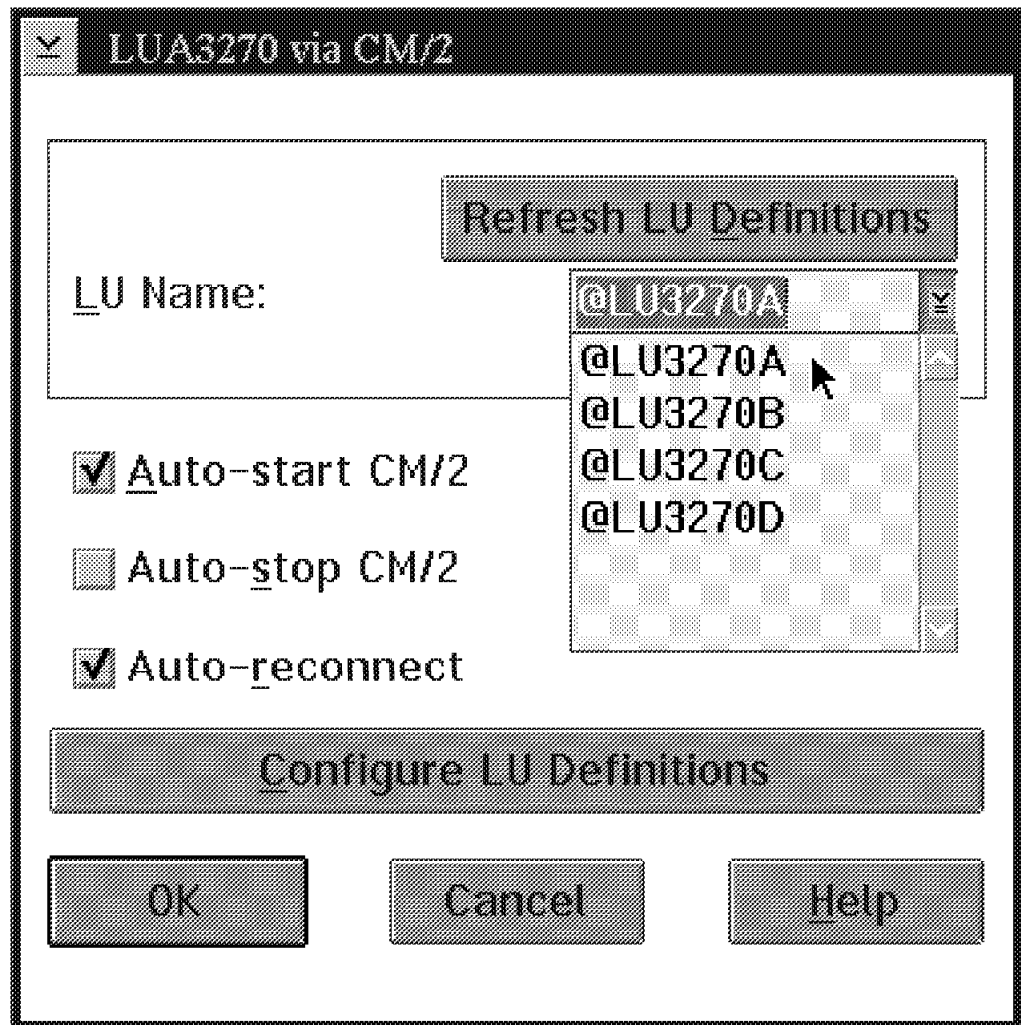


Figure 48. Selecting PCOMM via LU-A - Configuration

Note

If you do not see any LU names when configuring your PCOMM sessions, you should check your OS/2 Access Feature configuration. PCOMM requires that LU-A profiles must be defined first.

4.9 Monitoring the OS/2 Workstation

In most cases you will use the provided Subsystem Management utility program to monitor your OS/2 workstation running OS/2 Access Feature. There are also command line utilities you can invoke from the OS/2 command line, for example, CMQUERY, CMLINKS, DISPLAY and many others.

To check the correct operation of your connection select the Administration folder.

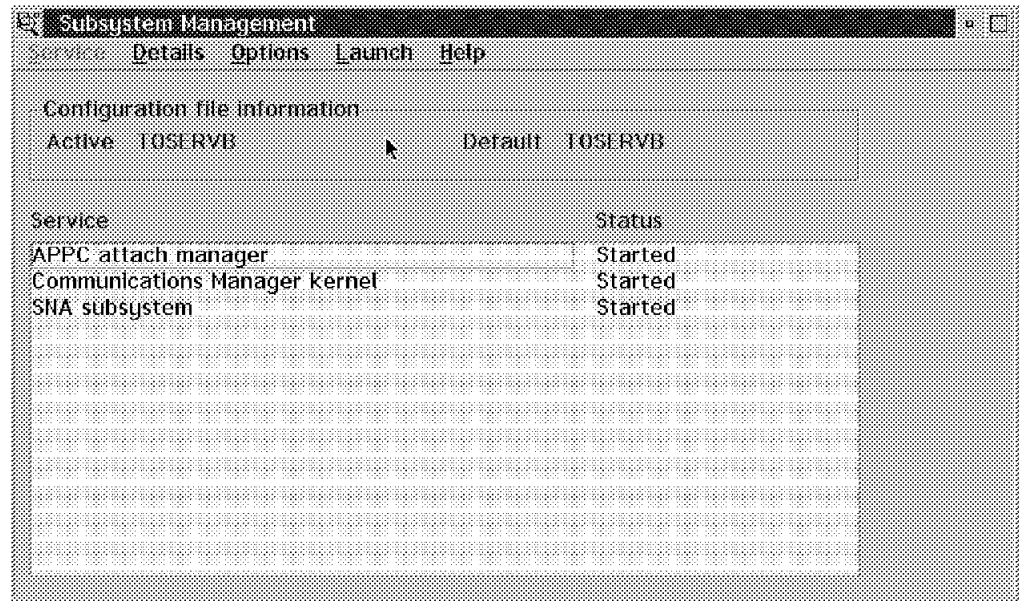


Figure 49. Monitoring OS/2 Access Feature

You can use the pull down menu in the Subsystem Management window (Figure 49) to start and stop the following services:

- APPC (Advanced Program-to-Program Communication) attach manager
- Communications Manager kernel
- SNA (Systems Network Architecture) subsystem

To start a service, select the service name, then select Service from the menu bar. Select Start from the Service pull down menu. When the service starts, the status field next to it changes to Started.

Note: You must start the SNA subsystem before you can start the APPC attach manager.

To stop a service, select the service name, then select Service from the menu bar. You have a choice of how to stop the service you selected. Select Stop Normal to stop the service when the current sessions end. Select Stop Abnormal to stop the service immediately.

It is a good idea to refresh the display in this window after you start or stop a service. When you refresh the display, you can be sure that the status column is reflecting the most current state of the service. To refresh the display, press F5 on your keyboard.

You can use the Subsystem Management window to display the status of individual communication sessions and the links on which they operate.

SNA Subsystems

To display the status of sessions and links on a Systems Network Architecture (SNA) network, select Details from the menu bar. Then select SNA subsystem from the Details pull down menu. The SNA Subsystem window appears, from which you can choose which resources you want to display or control.

In many cases, when you open a display window, you can use the pull down menu in the window to start, stop, or modify the activity of the individual resources displayed.

X.25 Physical Links

To display the status of physical links on an X.25 network, select Details from the menu bar. Then select X.25 physical links from the Details pull down menu.

When you open the link display, you can use the menu bar in the Display window to connect or disconnect individual links.

Session	Options	Help	LU Alias	Partner LU	Mode	Sessions
TOSRVBW2	USIBMRA	SERVERB	21	NN	2	Active

Figure 50. Monitoring Logical Links

Session	Options	Help	LU Alias	Partner LU	Mode	Sessions
wsos2	@000000		USIBMRA.SERVERB		CPSVCMG	2
wsos2	@000000		USIBMRA.SERVERB		SNASVCMG	1
wsos2	@000000		USIBMRA.SERVERB		INTER	1

Figure 51. Monitoring APPC Sessions

Opening Display Active Configuration, and selecting Display General SNA, Adapter you get the following, as shown in Figure 52 on page 83.

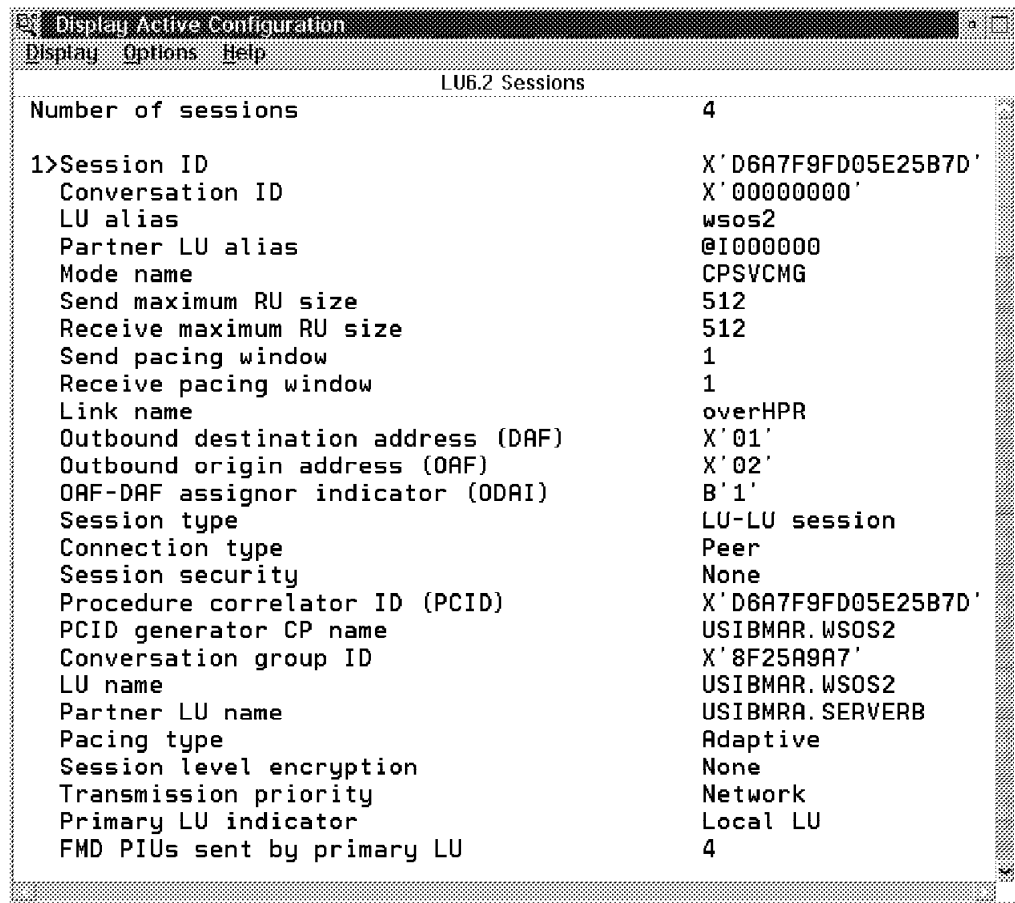


Figure 52. Displaying Session Details

You can use the Display Active Configuration window to display a variety of details about the configuration that is currently active on your workstation. Information is available for display in two general categories:

- General Systems Network Architecture (SNA)
 - Adapters
 - Global SNA
 - Incoming call answer status
 - Link definitions
 - Links
 - Logical units 0 to 3
 - Logical unit 6.2
 - LU definitions
 - Management services
 - Mode definitions
 - Partner LU definitions
 - Physical units
 - LU 6.2 sessions
 - SNA gateway

- System defaults
- Transaction program definitions
- Transaction programs
- X.25 physical links
- Advanced Peer-to-Peer Networking (APPN)
 - Class of service
 - Connection network
 - Directory
 - Intermediate sessions
 - Node
 - Topology

To display details about any of these resources, select Display from the menu bar. A pull-down menu appears from which you can select General SNA or APPN. When you select either of these options, a menu appears from which you can choose one of the resource types. When you select the item you want, Communications Manager displays details about the activity of that resource.

You can save the displayed information to a file by selecting Options from the menu bar, then selecting Display to File from the Options pull-down menu. By selecting one of the special file names for printers, you can direct the display output to your printer. To direct the display to your screen, select Options from the menu bar, then select Display to Screen.

You can also display the details of an active configuration on a remote workstation. Select Options from the menu bar, then select Select Target from the Options pull-down menu.

Note: You can also display this information by typing PMDSPLAY at your OS/2 command line.

Chapter 5. APPN and High-Performance Routing

In this chapter we present an overview of APPN and describe the position of APPN in relation to LEN and SNA, and introduce the basic terminology used with APPN. HPR architecture is then introduced in relation to the APPN architecture with basic operations and terminology explained. eNetwork Communications Server for OS/2 Warp Version 5.0 has enhanced its HPR implementation to support new architectures such as control flows over RTP connections and multilink transmission groups (MLTGs).

5.1 Overview

Advanced Peer-to-Peer Networking (APPN) is a networking extension to APPC which simplifies configuration and enhances management of computers using APPC or CPI-C transaction programs. An APPN network is made up of network nodes and end nodes. Nodes of other types can be connected to an APPN network node, and so use services of the APPN network. The APPN network nodes provide network-related services to network resources on network nodes and end nodes. An end node needs only know the location of its network node providing these services to communicate with any node in the APPN network.

SNA discovery service further simplifies configuration in APPN networks by automatically finding network nodes that are able to provide APPN services for the end nodes. Discovery service is a LAN address resolution protocol that can be used by a node on the LAN to find another node that matches given search criteria. Using this service a node can search for APPN network nodes, nodes that provide SNA boundary function, AS/400s, SNA gateways, or any user-defined class of servers. A node configured as a discovery server can respond to requests from clients as an APPN network node, a PU Type 2.0 gateway, or as a user-defined class of server.

For a more complete and detailed description of APPN functions and components see *APPN Architecture and Product Implementations Tutorial*, GG24-3669.

High-performance routing or HPR is an addition to the APPN architecture. It enhances APPN data routing performance and reliability, especially when using high-speed links. HPR provides non-disruptive rerouting around network outages, efficient selective retransmission, and end-to-end data integrity and congestion control.

To support emerging high-speed communications facilities, certain changes to the APPN architecture are required to allow switching in intermediate nodes to be done at a lower layer and much faster than can be achieved in base APPN. HPR changes the existing intermediate session routing (ISR), which is done in APPN by using a routing algorithm that minimizes the storage and processing requirements in intermediate nodes. The level of error recovery that is done in APPN for the slower-speed lines used today is unnecessary for high-speed, more-reliable lines. HPR addresses this by reducing the amount of error recovery done on individual lines and instead providing an end-to-end level of error recovery. HPR also enhances APPN by implementing a non-disruptive path switch function, which can switch sessions around failed links or nodes.

One of the general design principles of HPR is that it should have functional equivalence with the base APPN architecture. It was also a requirement that a node that supports HPR should be able to interoperate with existing APPN nodes. These two features will result in a seamless migration to HPR from an installed APPN network.

5.2 LEN and APPN

A network can be very simple, for example, two PCs connected by a telephone line, as shown in the figure below.

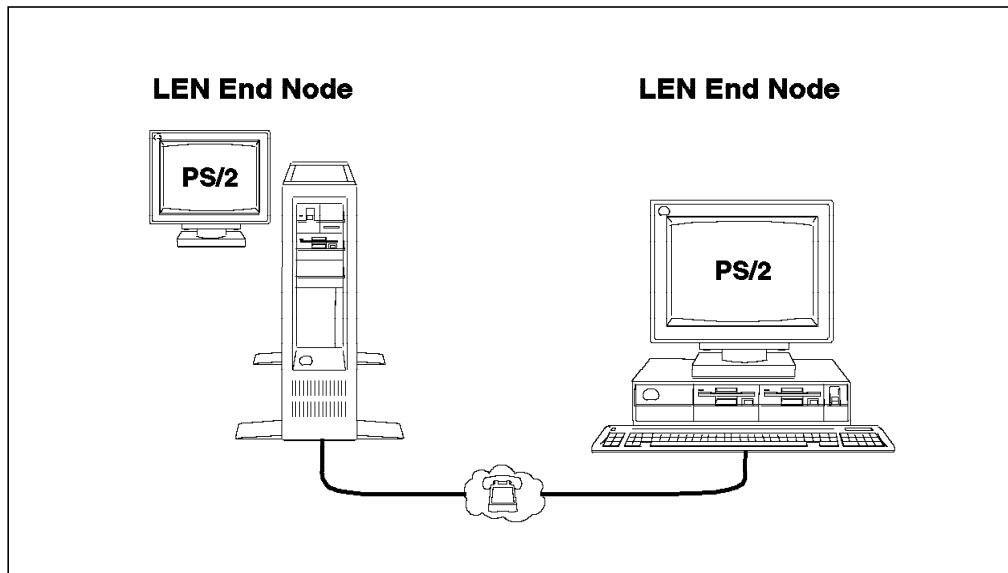


Figure 53. Two PCs Forming a LEN Connection

The purpose of connecting these two systems is to exchange data between two end users. An end user could be a person working with this system, a program running on the system, or a printer controlled by the system.

The end user gains access to the network through the logical unit (LU). Before the two LUs are able to exchange data, they must start an LU-LU session. For program-to-program communication, this session would typically be an LU 6.2 session.

In the case above, when the two systems (PCs) establish a *low-entry networking* (LEN) connection, the two connecting systems are known as *LEN end nodes*. Using the architectural terms, the configuration above could be drawn as shown in Figure 54 on page 89.

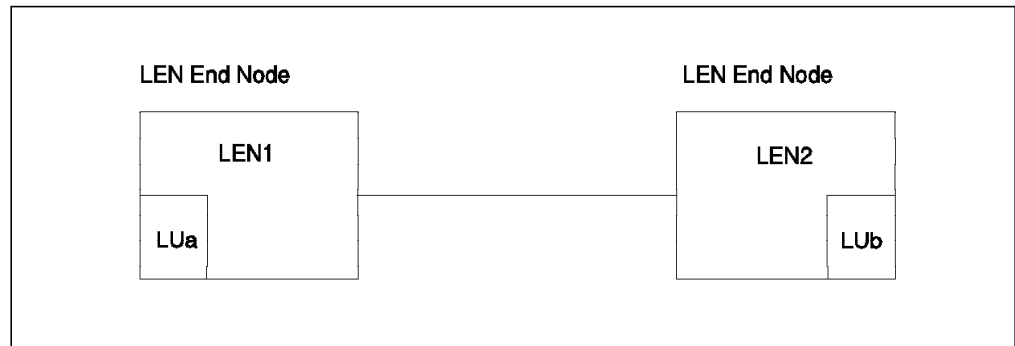


Figure 54. The Basic LEN Connection

Several systems can be configured as LEN end nodes, such as VTAM and NCP, AS/400 and PC. LEN end nodes provide the minimum functions required to:

- Provide a connection between LEN1 and LEN2
- Establish a session between the LUs named LUa and LUb
- Transport data

The relation between LEN end nodes is truly peer-to-peer. Either side may activate a connection or start a session to the partner.

A significant feature of the LEN architecture is that there are only *two* adjacent nodes involved in a LEN connection. No matter how many nodes there may be in the network, a LEN connection recognizes only two of them.

Obviously, there must be functions in addition to LEN if a network with more than two nodes is to be built. One of these functions is the capability to act as an intermediate node (that is, a node that can receive data that is not for itself and can pass it on to the destination node). This principle is shown in Figure 55.

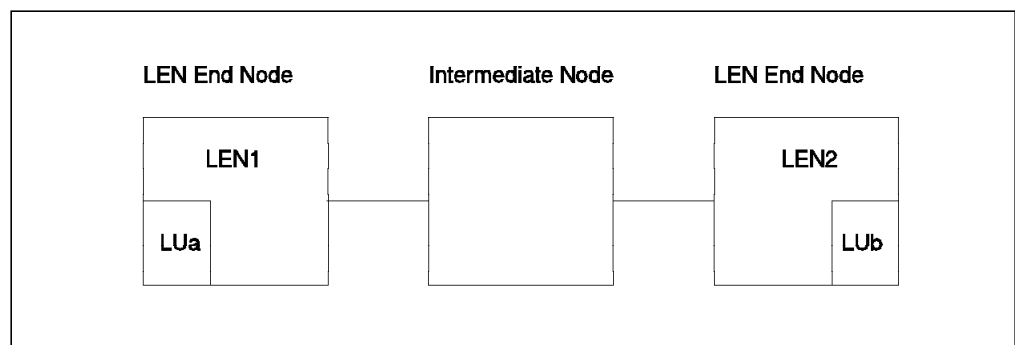


Figure 55. LEN End Nodes Connected to an Intermediate Node

According to the LEN architecture, the relation between LEN end nodes is always a “*two*-node peer relationship.” LUs residing on non-adjacent LEN nodes can establish sessions and exchange data because the intermediate node presents itself as a LEN node owning all LUs residing on non-adjacent nodes. As seen from LEN1, the intermediate node is just a normal LEN end node, and LEN2 is not visible at all from LEN1. For LEN1, the LU named LUb seems to be in the intermediate node.

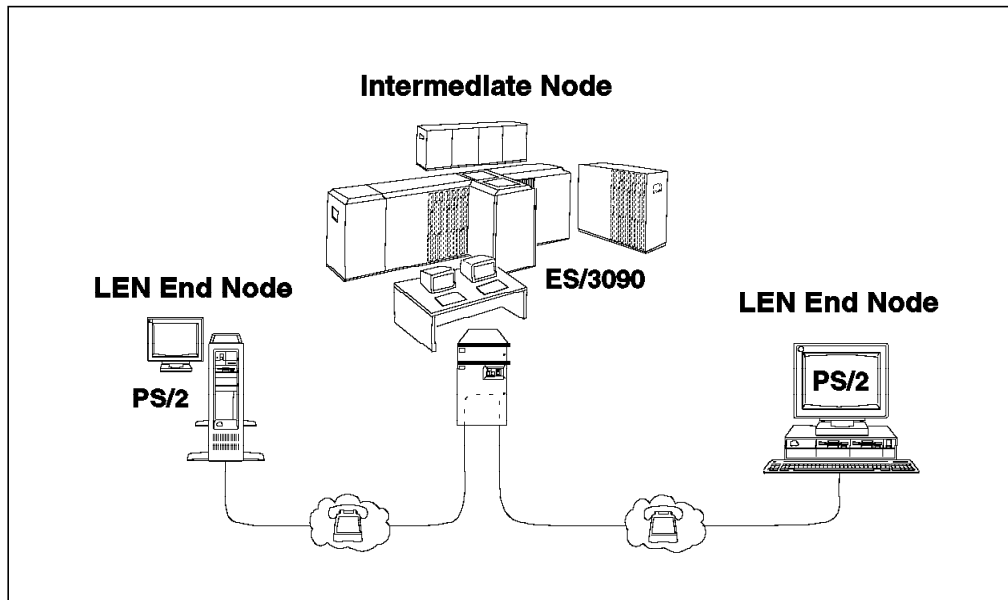


Figure 56. VTAM/NCP Providing the Intermediate Routing Function for LEN End Nodes

VTAM and NCP support the LEN end node function and also provide intermediate routing between LEN end node. Figure 56 gives an example of this configuration with VTAM on an ES/3090 as intermediate node.

The functions of LEN nodes are limited; for example, they are not able to exchange topology and configuration data. Additional functions are needed to reduce the number of definitions and the maintenance effort when building larger networks. For this purpose the APPN (APPN) architecture was developed and published as an extension to SNA (Systems Network Architecture).

APPN architecture defines two basic node types:

APPN End Node

The APPN end node is similar to a LEN end node, except that the control point (CP) of the end node exchanges information with the CP in the adjacent *network node*. The communication over the CP-CP sessions reduces the requirement for network definitions, and thus makes installation and maintenance of the network easier.

APPN Network Node

The APPN network node has intermediate routing functions and provides network services to either APPN or LEN end nodes that are attached to it. It establishes CP-CP sessions with its adjacent APPN network node to exchange network topology and resource information. CP-CP sessions between an APPN network node and an adjacent APPN end node are required only if the APPN end node is to receive network services (such as partner location) from the APPN network node.

APPN architecture also describes the connection of LEN end nodes to APPN network nodes or APPN end nodes.

Figure 57 on page 91 shows the basic form of an APPN network and gives an example of the services provided by the APPN network node. When LUa requests a session with LUC, the network node will locate the partner LU and assist in establishing the session.

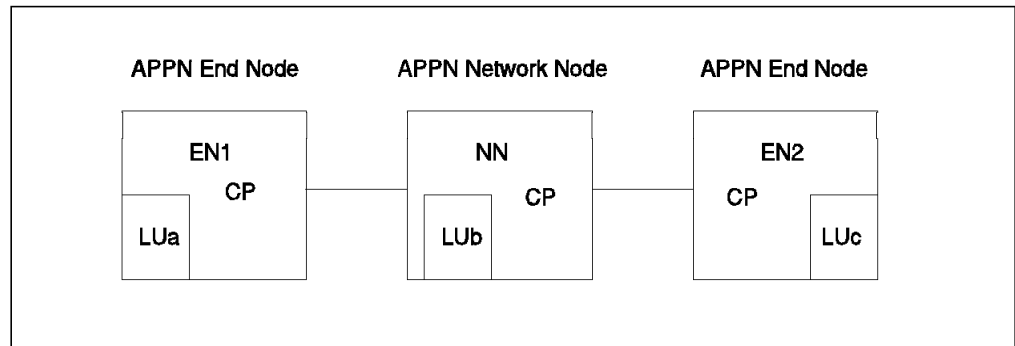


Figure 57. APPN with Three Nodes

Figure 57 shows the basic form of an APPN network. However, APPN networks can be much more complex. The architecture does not limit the number of nodes in an APPN network nor does it explicitly limit the number of intermediate APPN network nodes through which LU-LU sessions are routed. One restriction exists, however: the length of the Route Selection control vector (RSCV) describing a physical session path is limited to 255 bytes. See *Inside APPN - The Essential Guide to the Next-Generation, SG24-3669* for a detailed discussion.

Figure 58 shows a backbone structure of APPN network nodes to which end nodes connect. The APPN nodes communicate using CP-CP sessions between adjacent nodes. User sessions can be established from any LU to any LU.

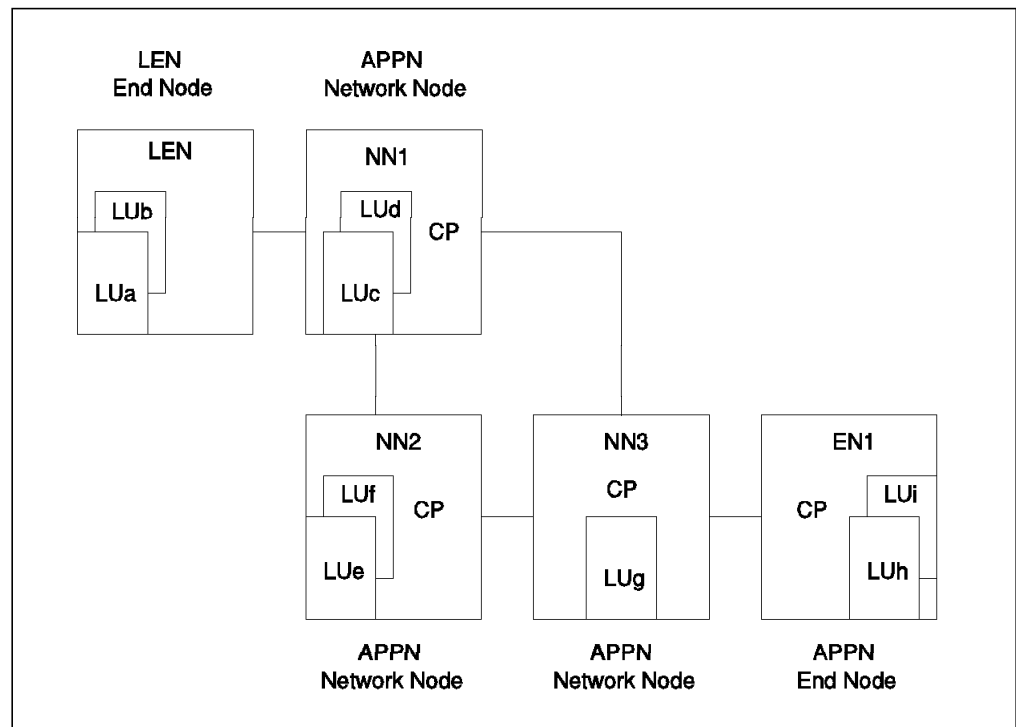


Figure 58. APPN Network with Different Node Types

While the previous figure showed the architectural node types used in the network, Figure 59 shows a variety of products, such as VTAM and NCP, AS/400, PC, and IBM 3174, connecting through different link protocols.

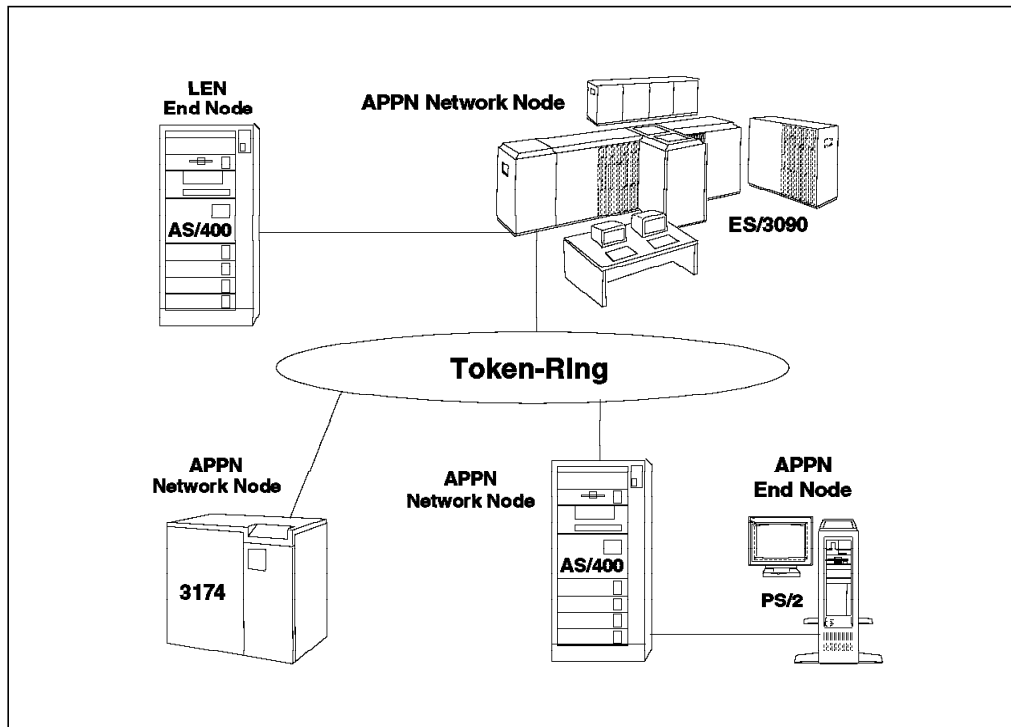


Figure 59. Advanced Peer-to-Peer Networking

Figure 59 depicts a VTAM host, an AS/400, and an IBM 3174 configured as APPN network nodes, a PC configured as an APPN end node and a second AS/400 configured as a LEN end node.

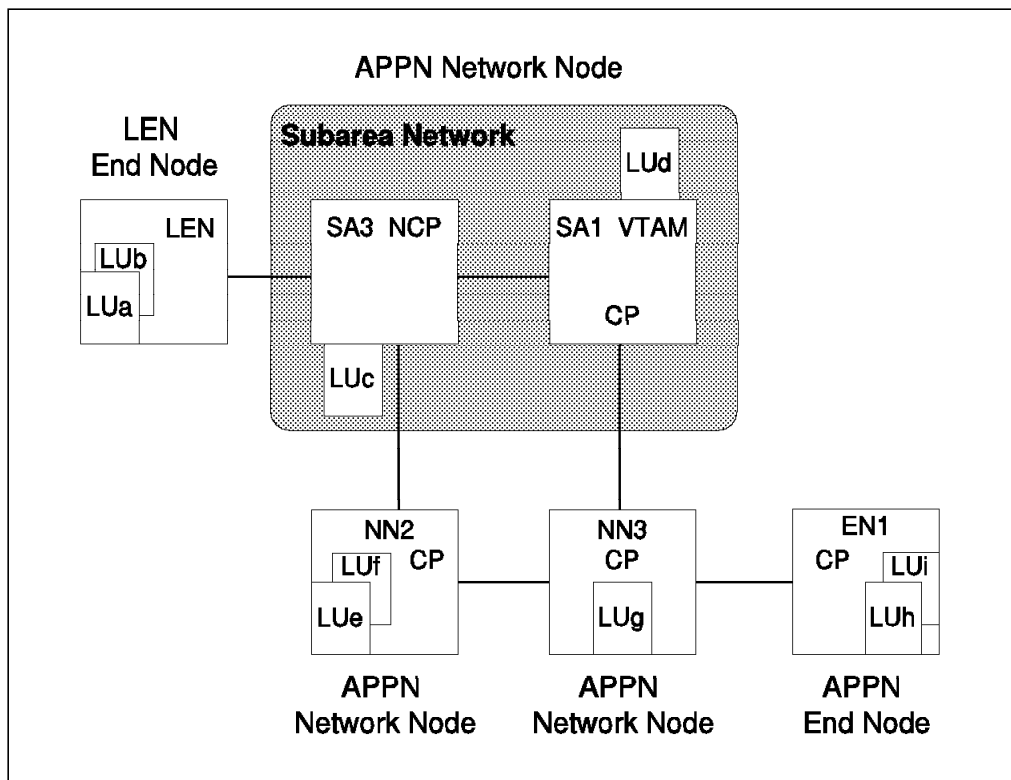


Figure 60. Composite Network Node with APPN Network Node Appearance

Note: A VTAM configured as a network node, together with all its owned NCPs, is called a *composite network node* (CNN). Within the composite network node, subarea protocols are used (see Figure 60) but to the other APPN or LEN nodes the CNN gives the appearance of a single APPN network node.

We have seen that the APPN architecture defines several types of nodes and that the CPs of these nodes have different functionality. The node types are defined more precisely later in this chapter. The CP functions are covered in several chapters in *Inside APPN - The Essential Guide to the Next-Generation*, SG24-3669. Internal implementations may be different in different products.

5.3 High-Performance Routing (HPR)

High-performance routing (HPR) is an extension to the APPN architecture. It can be implemented on an APPN network node or an APPN end node. It does not change the basic functions of the architecture.

IBM's intent is that installation of HPR will always be by software upgrades to existing APPN products, not hardware changes.

HPR enhances APPN routing mechanisms as follows:

- HPR improves performance, especially on high-speed links.
- It can nondisruptively route sessions around links or nodes that have failed.
- It provides a new mechanism for congestion control that can improve traffic throughput.
- It reduces the amount of storage required in APPN intermediate nodes.

5.4 Transmission Groups

The connections between APPN nodes are called *transmission groups* (TGs).

The base APPN architecture allows for *single-link transmission groups* only. APPN with High Performance Routing supports both single-link and *Multi-Link Transmission Groups* (MLTGs). See 5.11.4, "Multi-Link Transmission Groups" on page 123 for details of the MLTG protocol.

Note: Do not confuse Multi-Link Transmission Groups with *parallel* transmission groups. A multi-link TG consists of multiple DLC-level connections between two nodes made to appear to higher layers as a *single* connection. The essential purpose of this is to have one "link" between the nodes that is better than the component links individually (typically in bandwidth, or availability, or both). Parallel transmission groups, on the other hand, comprise several links or several groups of links designed to appear to higher layers as *multiple* connections between the nodes. Their essential purpose is to augment the pool of possible routes around a network.

When a node has TGs connecting to more than one other node, it has *multiple* TGs.

5.5 Names

Resource naming is important as it allows end users to start sessions without knowing the locations of other resources in the network.

5.5.1 The Network Accessible Unit

In an APPN network, all components that can establish sessions with one another are called network accessible units. Examples are CPs and LUs. The term NAU was previously used as an abbreviation for “network *addressable* unit.” The terminology has changed with APPN. Now NAUs are represented by names rather than by addresses.

Note: NAU names must be unique within an APPN network. To ensure this uniqueness you need a consistent naming convention.

5.5.2 Network Identifiers

You can divide your “network” into partitions in order to simplify your resource name administration. Each partition will have a network identifier (network ID), 1 to 8 bytes long. Net IDs are used throughout SNA, in both subarea and APPN parts of networks. Because names of LUs and CPs have to be unique only within the scope of a network ID, you can assign and administer them independently for each partition.

Registering can help your network administrators ensure the uniqueness of a network ID. IBM provides a worldwide registry for network IDs. Information on the registration process can be obtained from your IBM representative.

IBM-registered network IDs should have an 8-character name of the form cceeeenn, where:

cc is the country code (according to ISO Standard 3166).

eeee is the enterprise code (unique within a country).

nn is the network suffix code (unique within one enterprise).

5.5.3 Network Names

A *network name* is an identifier of a network resource. Each CP, LU, link, and link station in an SNA network has a network name. The network names are assigned through system definition. In an APPN node, the system definition is done using the node operator facility (NOF).

5.5.4 Network-Qualified Names

A resource’s *network-qualified name* identifies both the resource and the network in which the resource is located. It is a concatenation of the network ID and the network name of the resource. For example, NETA.LUA, NETA.LUB, NETB.LUA, and NETB.LUB are all valid network-qualified names, and they refer to four different entities.

5.6 Addresses

Addresses are used in all SNA networks for routing data correctly between session partners. There are big differences, however, in the ways addresses are used in traditional subarea SNA on the one hand and APPN on the other, and differences again between basic APPN and HPR.

5.6.1 Addresses in Subarea Networks

In traditional subarea SNA, each resource is assigned its own distinct network address. The subarea number part of this is used by VTAM and NCP nodes in the network to route data to the correct destination subarea. There, local addressing takes over. The boundary function of the VTAM or NCP node concerned converts the network addresses to local addresses. These are seen in the transmission headers of packets on boundary links.

5.6.2 Addresses in APPN Networks

In an APPN network, routing information is session oriented throughout. The *address* used in an APPN transmission header is an identifier unique on the given TG for a particular session, rather than the address of the NAU. The identifiers are locally defined for each pair of adjacent routing nodes and are only *temporarily* assigned. They are assigned at session initiation, and released when the session ends. The session initiation request (BIND) carries routing information about the full session path that determines the sequence of links used from origin to destination. The local session identifier stored in each intermediate node in a session path is contained in a *session connector* and kept only for the life of the session.

The session identifier is associated with:

- A particular session
- A transmission group between two nodes

Figure 61 shows a session between two LUs, LUa and LUb, residing on two non-adjacent APPN end nodes. The session data is routed through two intermediate network nodes. The session can be thought of as a sequence of three session stages or *hops* with a distinct session identifier assigned to each session stage.

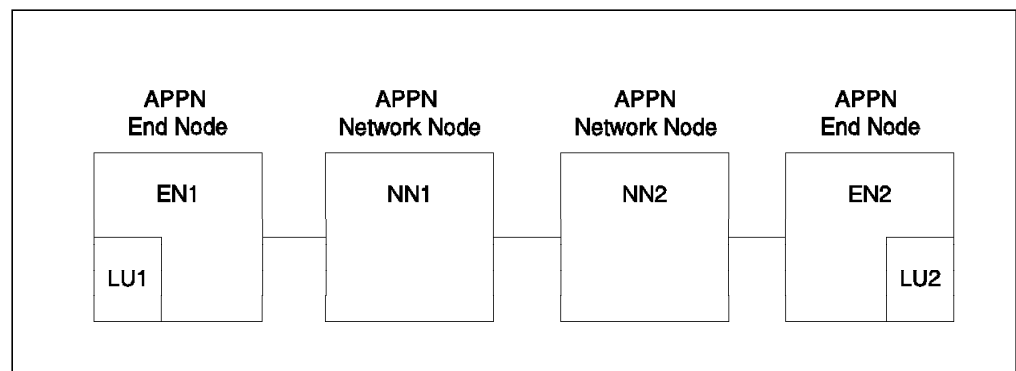


Figure 61. Session with Several Session Stages

Session identifiers vary at different session stages, which is why they are called *local-form session identifiers* (LFSIDs). The LFSID is set up during session establishment by the address space manager component of the CP and assigned

for the lifetime of an LU-LU (or CP-CP) session. Details may be found in *Inside APPN - The Essential Guide to the Next-Generation*, SG24-3669.

Each session is uniquely identified by a network-unique identifier, the Fully Qualified Procedure Correlation Identifier (FQPCID), which is described in *Inside APPN - The Essential Guide to the Next-Generation*, SG24-3669.

5.6.3 Addresses in HPR Networks

In an HPR network, a new form of routing is used, which is called *automatic network routing* (ANR). ANR is a source-routing protocol, which means the sender of a packet provides the information about the physical path the packet will use through the network in the network header. As HPR provides the ability to do non-disruptive path switching, the HPR architecture handles the case where the route changes in mid-session.

ANR uses a new form of addressing to identify the route through an HPR network. However, unlike the APPN session-oriented addresses (LFSIDs), the addresses in ANR are based purely on the TGs that make up the route. The network header contains a list of ANR labels that identify the route through the network. Each ANR label describes a TG that is to be taken to exit a node. This is described in more detail in *Inside APPN - The Essential Guide to the Next-Generation*, SG24-3669.

In addition to the ANR labels, there are still addresses that are associated with sessions in HPR. Each session will have a pair of unique session addresses, one for each direction. Unlike the LFSID that identifies each stage of the APPN session, the HPR session addresses are used only on an (HPR) end-to-end basis. They are known as *enhanced session addresses*.

The process of supporting the end-to-end sessions across the HPR network is called rapid transport protocol or RTP. This is described in more detail in *Inside APPN - The Essential Guide to the Next-Generation*, SG24-3669.

In a network that is supporting both existing APPN nodes and HPR nodes, both the APPN and the HPR methods of addressing are used. This is described in more detail in *Inside APPN - The Essential Guide to the Next-Generation*, SG24-3669.

5.7 Domains

A domain is an *area of control*. A domain in an APPN network consists of the control point in a node and the resources controlled by the control point. Consequently, all APPN networks are multidomain networks.

Though all APPN nodes are peers with respect to session initiations and do not rely on other nodes to control their resources, APPN end nodes and LEN end nodes do use the services of network nodes. The domain of an APPN end node or LEN end nodes contains the node's own (local) resources. The domain of an APPN network node contains its local resources *and* the resources of those nodes that use the network node's services. Thus, the domains of the APPN end nodes and LEN end nodes are included in the domains of their respective network node servers.

Note: In traditional subarea networking, a domain is the part of the network owned by a VTAM System Services Control Point (SSCP). Within this document,

when using the term domain, we refer to an APPN domain unless explicitly stated otherwise.

5.7.1 APPN Network Node

An APPN network node provides distributed directory and routing services for all LUs that it controls. These LUs may be located on the APPN network node itself or on one of the adjacent LEN or APPN end nodes for which the APPN network node provides network node services. Jointly, with the other active APPN network nodes, an APPN network node is able to locate all destination LUs known in the network.

A facility known as *central resource registration* allows an APPN network node to register its resources at a central directory server. Once a resource is registered, APPN network nodes can locate the resource by querying the central directory server instead of using a broadcast search, thus improving network search performance during session establishment.

After the LU is located, the APPN network node is able to calculate the route between origin and destination LU according to the required class of service. All network nodes exchange information about the topology of the network. When two adjacent network nodes establish a connection, they exchange information about the network topology as they know it. In turn, each network node broadcasts this network topology information to other network nodes with which it has CP-CP sessions.

Alternatively, if the connection between network nodes is deactivated, then each network node broadcasts this change to all other active adjacent network nodes. An APPN network node that is taken out of service will be declared inactive and, after some time, removed from the topology information in all network nodes together with its routing capabilities to other nodes.

The APPN network node is also capable of routing LU-LU sessions through itself from one adjacent node to another adjacent node. This function is called intermediate session routing.

5.7.2 APPN End Node

An APPN end node provides limited directory and routing services for its local LUs. It can select an adjacent APPN network node and request this network node to be its *network node server*. If accepted by the network node, the APPN end node may register its local resources at the network node server. This allows the network node server to intercept Locate search requests for the APPN end node's resources and pass these requests to the APPN end node for verification.

Without a network node server an APPN end node can function as a LEN end node and establish LU-LU sessions with a partner LU in an adjacent APPN or LEN node.

When it needs to find an LU it does not already know, an APPN end node sends a Locate search request to its network node server. The network node server uses its distributed directory and routing facilities to locate the LU (via directed, central directory, or broadcast searches) and calculates the optimal route to the destination LU from the APPN end node.

The APPN end node may have active connections to multiple adjacent network nodes. At any given moment, however, only one of the network nodes can be acting as its network node server. The APPN end node establishes CP-CP sessions with a network node to select that network node as its network node server.

On APPN network nodes, APPN end nodes are categorized as either *authorized* or *unauthorized*. An authorized APPN end node may send registration requests to register local network accessible resources at a network node server, a facility known as *end node resource registration*, and may, in addition, request that these resources be registered with the central directory server. If during session establishment a network node server does not know where an LU is located, it will query authorized APPN end nodes within its domain that have indicated they are willing to be queried for unknown resources. Network accessible resources on unauthorized nodes require explicit definition at the network node server, either statically as part of its system definition, or dynamically by the network node server's operator. To avoid unnecessary explicit definitions of resources of authorized APPN end nodes at their network node servers, you should have them register their resources, or be set up to allow the network node servers to query them for unknown resources.

An APPN end node can attach to any LEN or APPN node regardless of its network ID.

5.7.3 LEN End Node

A LEN end node provides peer-to-peer connectivity to other LEN end nodes, APPN end nodes, or APPN network nodes. A LEN end node requires that all network accessible resources, either controlled by the LEN end node itself or on other nodes, be defined at the LEN end node. LUs on adjacent nodes need to be defined with the control point name of the adjacent node. LUs on non-adjacent nodes need to be defined with the control point name of an adjacent network node, as LEN end nodes assume that LUs are either local or reside on adjacent nodes.

Unlike APPN end nodes, the LEN end node cannot establish CP-CP sessions with an APPN network node. A LEN end node therefore cannot register resources at a network node server. Nor can it request a network node server to search for a resource, or to calculate the route between itself and the node containing a destination resource. It does, however, use the distributed directory and routing services of an adjacent network node indirectly. It does this by predefining remote LUs, owned by non-adjacent nodes, with the CP name of an adjacent APPN network node. The session activation (BIND) request for that remote LU is sent by the LEN end node to the adjacent network node. The network node, in turn, automatically acts as the LEN end node's network node server, locates the actual destination LU, calculates the route to it, and uses this route to send the BIND.

A LEN end node can attach to any LEN or APPN node regardless of its network ID.

5.7.4 Branch Extender Node (BrNN)

The branch extender node is a border node subset that is designed to interconnect a branch office consisting of LANs, end nodes and LEN end nodes with dependent and independent LUs and PUs such as teller machines to an APPN WAN backbone network. The branch extender function provides optimized peer-to-peer communication, so large APPN network customers can connect many LAN-based branches via one large APPN WAN primarily based on switched links. It addresses having a very large number of network nodes in the topology database. It adds the ability to register resources from the Branch Extender node to a central directory server in the WAN. In conjunction with APPN switched connection networks, it also makes it possible to have direct single-hop connections between two APPN nodes in different APPN subnetworks that are in the same switched connection network.

A branch extender node is a network node that typically has both LAN and WAN interfaces. To domain devices (downlinks, typically LAN), a branch extender node looks like an APPN network node. To uplink devices a branch extender node looks like an APPN end node, so it really appears as a network node or an end node depending upon where the viewer is in the network. For more information about branch extender nodes (BrNN) see Chapter 7, "Branch Extender Node" on page 157.

5.7.5 Other Node Types

In SNA, a node represents an endpoint of a link or a junction common to two or more links. The LEN end node, APPN end node, and APPN network node are endpoints of a link. Each node has a distinct role in an APPN network.

Besides these node types you will find references in the APPN literature to other node types that are either synonyms for nodes as seen from a subarea network, represent a specific junction in the network, or represent an APPN node with additional functions. The following is not a complete list, but it does contain all types found when creating this document:

- Boundary and peripheral node
- Composite node
- Interchange node
- Virtual Routing node
- Peripheral border node
- Extended border node
- HPR node

5.7.5.1 Boundary and Peripheral Node

In traditional subarea SNA networks, resources are controlled through hierarchical structures. Nodes in these networks are categorized as *subarea* and *peripheral* nodes. An example of such an SNA network is an IBM System/390 mainframe running VTAM with a 3745 communication controller running NCP, and 3270 terminals attached via IBM 3274 controllers. The VTAM and NCP nodes are both referred to as subarea nodes. The VTAM subarea node contains the System Services Control Point (SSCP). Like the APPN control point, the SSCP controls all the resources in its domain.

Attached to these subarea, or *boundary*, nodes are the peripheral nodes. The peripheral node is either a PU T2.0 or an APPN or LEN node. The PU T2.0 node, for instance one of the IBM 3274 clusters in our example network, is a traditional

hierarchical node that requires the support of an SSCP to establish sessions, and of the boundary function for its addressing.

Traditional subarea SNA allowed LEN connections only; CP-CP sessions could not be established between VTAM and the APPN nodes.

With the introduction of APPN VTAM, a VTAM or a composite network node (subarea network consisting of one VTAM and one or more NCPs) is able to present an APPN image to other APPN nodes. APPN VTAM allows CP-CP sessions with APPN nodes attached to the VTAM or NCP boundary function to gain full APPN connectivity. The term *peripheral* node has lost its value in a network that is truly peer-to-peer.

5.7.5.2 Composite Node

The term *composite node* is used in some publications to represent a group of nodes that appear as one APPN or LEN node to other nodes in an APPN network. For example, a subarea network consisting of a VTAM host and some NCPs is a multiple-node network, but when connected to an APPN node, appears as *one* logical APPN or LEN node.

A subarea composite node may appear as either a LEN end node or as an APPN network node. In the former case, the term composite LEN node is used; in the latter case the term composite network node (CNN) is used.

5.7.5.3 Interchange Node

A VTAM host acting as an interchange node (ICN) can be a stand-alone APPN VTAM node or a composite network node. The ICN routes sessions from APPN nodes into and through the subarea network using subarea routing, without exposing the subarea implementation to the APPN part of the network. This is accomplished by making the APPN VTAM node, plus all its owned resources, appear to other nodes as a single APPN network node with multiple connections. At the same time the ICN, and the NCPs it owns, will maintain their subarea appearance to other subarea nodes.

The ICN supports SSCP-SSCP sessions with other VTAM nodes as well as CP-CP sessions with adjacent APPN network nodes and end nodes. This support allows the ICN to use both APPN and subarea data flows to locate LUs and to provide the best route between nodes. APPN session setup protocols, which flow on CP-CP sessions, are converted to the corresponding subarea protocols that flow on SSCP-SSCP sessions, and vice versa.

To an ICN, see for example VTAM1/NCP in Figure 62 on page 101, multiple VTAMs and NCPs may connect using subarea protocols. Session establishment is possible between any LU in the subarea network and any LU in the APPN network. The VTAM host to which APPN nodes attach, or the VTAM host owning the NCPs to which APPN nodes attach, must have implemented APPN VTAM, as it is responsible (as an *interchange node*) for the conversion of subarea to APPN protocols and vice versa. Other VTAMs within the subarea network may be backlevel VTAMs. From the viewpoint of the APPN nodes, LUs owned by VTAMs (for example, VTAM2 or VTAM3) other than the VTAM providing the interchange function are considered to reside on APPN end nodes.

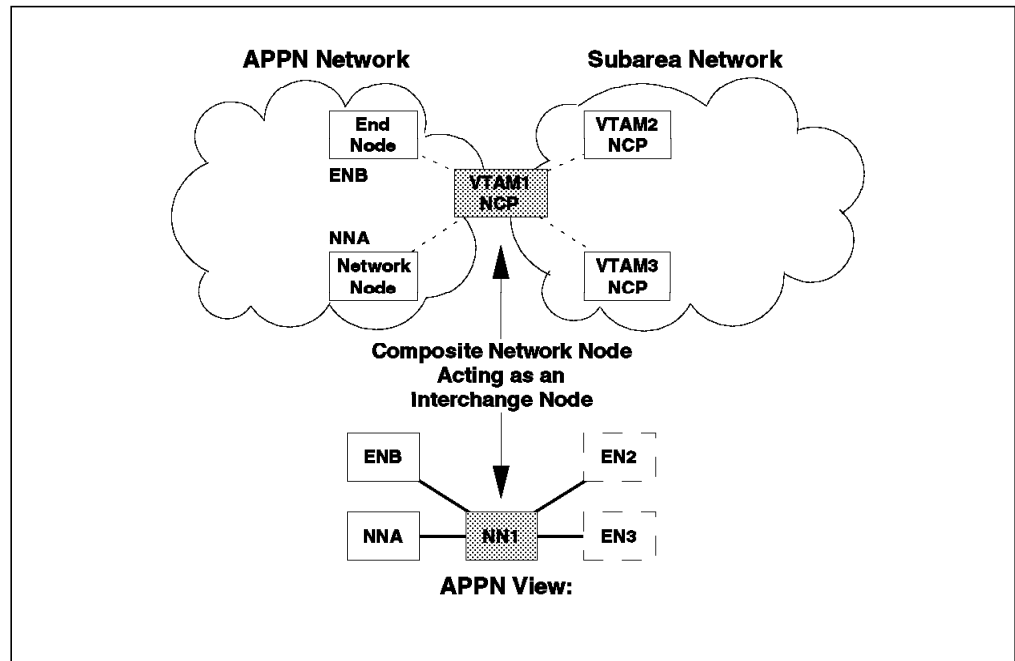


Figure 62. Composite Network Node Acting as an Interchange Node

Note: Figure 62 shows the basic form of connecting APPN and subarea networks using a composite network node acting as an interchange node. For more details see *Inside APPN - The Essential Guide to the Next-Generation*, SG24-3669.

5.7.5.4 Virtual Routing Node / Connection Networks

APPN allows APPN nodes to reduce the addressing information stored at each node connected to a shared-access transmission facility (SATF), such as a token-ring, by allowing each node to define a virtual routing node (VRN) to represent its connection to the shared facility and all other nodes similarly configured. The SATF and the set of nodes having defined a connection to a common virtual routing node are said to comprise a *connection network*.

A virtual routing node (VRN) is not a node, but it is a way to define an APPN node's attachment to a shared-access transport facility. It reduces end node definition requirements by relying on the network node server to discover the common connection and supply necessary link-level signaling information as part of the regular Locate search process. LU-LU session data can then be routed directly, without intermediate node routing, between APPN nodes attached to the SATF. For more information see *Inside APPN - The Essential Guide to the Next-Generation*, SG24-3669.

5.7.5.5 Border Node

Base APPN architecture does not allow two adjacent APPN network nodes to connect and establish CP-CP sessions when they do not have the same net ID. The border node is an optional feature of an APPN network node that overcomes this restriction.

A border node can connect to an APPN network node with a different net ID, establish CP-CP sessions with it, and allow session establishment between LUs in different net ID *subnetworks*. Topology information is not passed between the subnetworks. Similarly a border node can also connect to another border node.

Two types of border node are defined in the APPN architecture: peripheral border node and extended border node.

Note

eNetwork Communications Server for OS/2 Warp cannot be configured as a border node.

5.7.5.6 Peripheral Border Node

The peripheral border node enables the connection of network nodes with different net IDs and allows session establishment between LUs in different, adjacent, subnetworks.

A peripheral border node provides directory, session setup and route selection services across the boundary between paired subnetworks with different net IDs while isolating each subnetwork from the other network's topology information. This reduces the flow of topology updates and the storage requirements for the network topology database on network nodes in each of the network partitions.

Note

eNetwork Communications Server for OS/2 Warp cannot be configured as a peripheral border node.

5.7.5.7 Extended Border Node

The extended border node allows the connection of network nodes with different net IDs, and session establishment between LUs in different net ID subnetworks that need not be adjacent.

An extended border node provides directory, session setup and route selection services across the boundary between paired or cascaded non-native net ID subnetworks. An extended border node can also partition a single net ID subnetwork into two or more clusters or topology subnetworks with the same net ID, thus isolating one from the topology of the other.

Note

eNetwork Communications Server for OS/2 Warp cannot be configured as an extended border node.

5.7.5.8 HPR Node

An HPR node is an APPN node that has implemented the optional HPR functions. An HPR node can be an APPN end node or an APPN network node.

In a mixed APPN and HPR topology network, a group of interconnected HPR nodes is sometimes referred to as an *HPR subnetwork* or an HPR subnet. When an HPR link is activated between a pair of adjacent HPR nodes, an HPR subnet is formed.

In addition, the terms base APPN subnetwork and base APPN subnet may also be used when referring to a part of the network that is not an HPR subnet. HPR subnets are not separated from the other parts of the topology database.

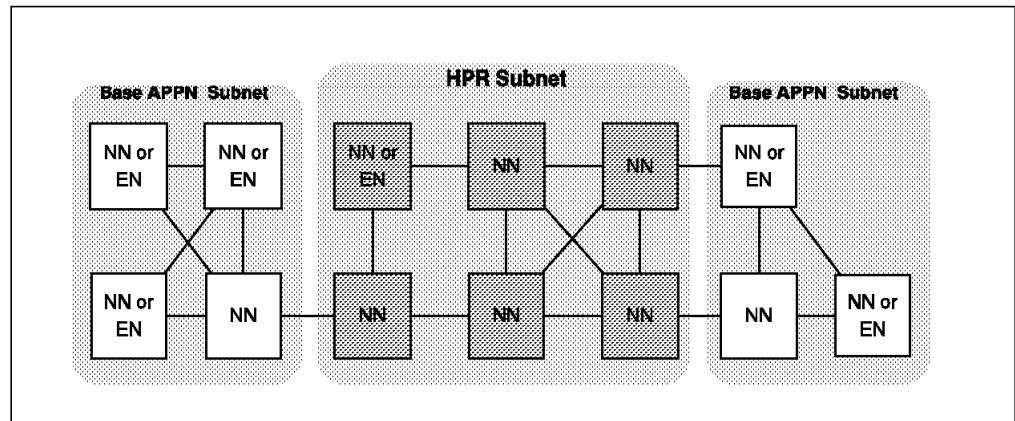


Figure 63. HPR Nodes and HPR Subnets

Figure 63 shows a backbone HPR subnet with two adjacent base APPN subnets. The six nodes in the HPR subnet are interconnected with HPR links.

If any of the nodes are to provide intermediate session routing, then they must be network nodes. But if a node acts only as a session endpoint, it can be a network node or an end node. The HPR nodes are exactly the same as APPN nodes in this respect.

If a product supports HPR, it can choose to implement only the base HPR function, or the HPR base function and optional functions. The base HPR function provides ANR routing, so as a minimum an HPR node can always act as an intermediate node in an HPR network. An HPR node that is providing only ANR routing will always be a network node. For more information, see *Inside APPN - The Essential Guide to the Next-Generation*, SG24-3669.

5.8 RTP and ANR

The two main components of HPR are the rapid-transport protocol and automatic network routing.

5.8.1 Rapid Transport Protocol (RTP)

RTP is a connection-oriented, full-duplex protocol designed to support data in high-speed networks. *RTP connections* are established within an HPR subnet and are used to carry session traffic. These connections can be thought of as *transport pipes* over which sessions are carried. RTP connections can carry data at very high speeds by using low-level intermediate routing and minimizing the number of flows over the links for error recovery and flow control.

The RTP functions include:

Non-disruptive path switch

An RTP connection's physical path can be switched automatically to reroute sessions around a failure in the network. The RTP connection is reestablished over a new physical path that bypasses the failing link or node, and the session's traffic flow is resumed on the RTP connection non-disruptively. Any data that was in the network at the time of the failure will be recovered automatically using RTP's end-to-end error recovery.

End-to-end error recovery

In base APPN, error recovery is done on every link in a network. To address the emerging high-speed lines with lower bit error rates, HPR removes the requirement to do link-level error recovery and instead does error recovery on an end-to-end basis. This will improve performance by reducing the number of flows required to do the link-level error recovery on every link. RTP also supports selective retransmission, where only missing or corrupted packets are re-sent, and not all packets, since the failure occurred.

RTP also handles the in-order delivery of data. If there is a Multi-Link Transmission Group (MLTG) in the path, the packets may arrive at the endpoint out of sequence. The RTP endpoints will re-sequence the data in this case.

End-to-end flow control and congestion control

Flow control in APPN networks is also done on each stage of the session by using adaptive session-level pacing. This method provides the best performance for networks comprised of a mixture of link types, with differing speeds and quality. However, for high-speed networks, adaptive session-level pacing is not adequate because of the amount of processing required in each node.

HPR uses a protocol suitable for high-speed routing called adaptive rate-based (ARB) flow/congestion control. It regulates the flow of traffic over an RTP connection by adaptively changing the sender's rate based on feedback from the receiver. This protocol allows for high link utilization and prevents congestion before it occurs, rather than recovering from congestion once it occurs.

Figure 64 shows an RTP connection that is carrying multiple sessions. Traffic from many sessions requesting the same class of service can be routed over the same RTP connection. If an HPR node is an intermediate node on a session path, then it must be a network node, just as in base APPN.

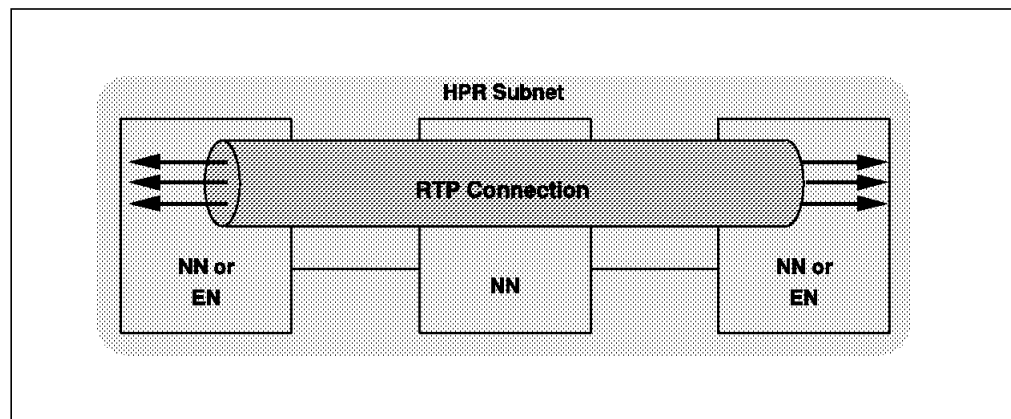


Figure 64. RTP Connection Supporting APPN Sessions. Multiple sessions using the same class of service can share the connection.

5.8.2 Automatic Network Routing (ANR)

ANR is a new routing mechanism to minimize storage and processing requirements for routing packets through intermediate nodes.

The ANR functions include:

Fast packet switching

ANR takes place at a lower layer than APPN intermediate session routing and will significantly improve performance in the intermediate nodes. Functions such as link-level error recovery, segmentation, flow control, and congestion control are no longer performed in the intermediate nodes. Instead, these functions are performed at the RTP connection endpoints.

No session awareness

Intermediate nodes are not aware of the SNA sessions or the RTP connections that are established across the nodes. This means that there is no requirement to keep the routing tables for session connectors that are kept in base APPN (APPN sessions require between 200 and 300 bytes per session per node). This saving on intermediate storage will be essential in the future, when HPR nodes supporting high-speed links will be carrying many more intermediate sessions than APPN nodes do today.

Source routing

ANR is a source-routing protocol and carries the routing information for each packet in a network header with the packet. Each node strips off the information it has used in the packet header before forwarding onto the link, so the next node can easily find its routing information at a fixed place in the header. This means that switching packets through a node can be done more quickly than in the routing table lookup method used in base APPN. There is no restriction on the number of hops in ANR.

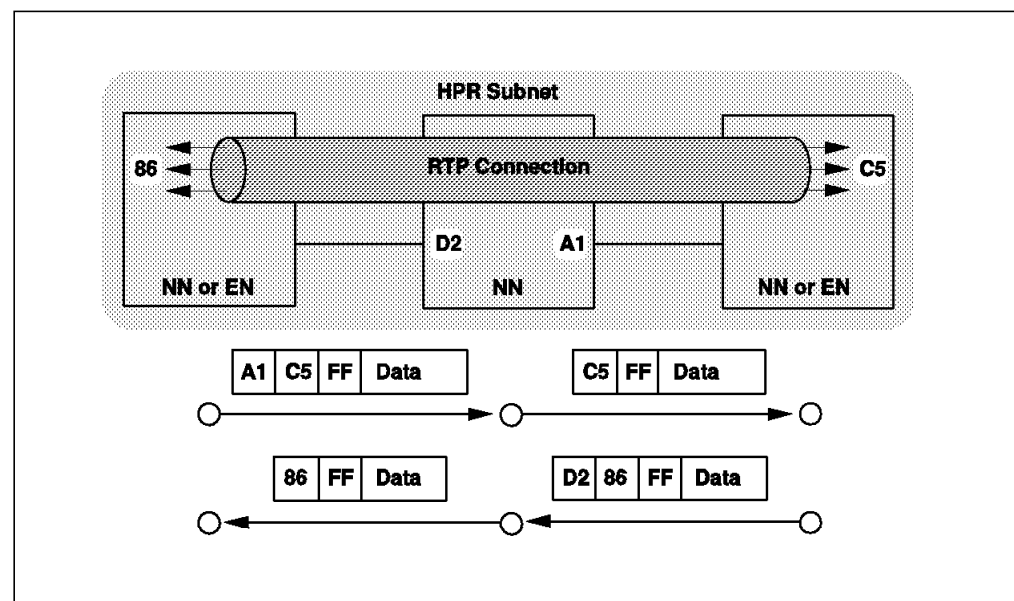


Figure 65. ANR Routing. Intermediate nodes strip routing information from the header at every stage through the network.

Figure 65 shows the principle of ANR. The intermediate network node strips the first routing label (A1) from the network header before forwarding the packet on link A1. The address of C5 represents the endpoint in the last HPR node. The intermediate network node can route packets very quickly, with no need to reserve storage or buffers, or to do link-level error recovery.

5.8.3 General APPN/HPR Operation

By way of a general overview, an example of the setting up and operation of an APPN session that passes through an HPR subnet is given here. The details of the HPR parts of the route and session setup are explained in the later parts of this chapter.

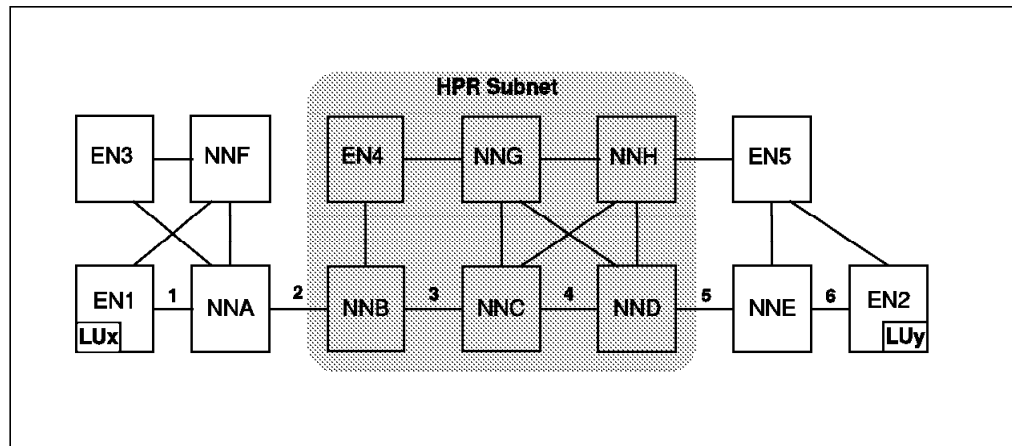


Figure 66. Overview of the Operation of an APPN/HPR Network. NNA is the network node server for EN1 and NNE is the network node server for EN2. The assumption is that LUx on EN1 wants to establish a session with LUy on EN2.

5.8.3.1 APPN Topology

CP-CP sessions are established between adjacent nodes as in base APPN. The CP-CP sessions between the network nodes are used to broadcast the topology database updates. In the example, it is assumed that the CP-CP sessions between network nodes are fully meshed and so there will be CP-CP sessions between the following pairs of network nodes:

NNA-NNB, NNB-NNC, NNC-NND, NND-NNE, NNG-NNH, NNA-NNF, NNC-NNG, NNC-NNH, NND-NNG and NND-NNH.

The APPN topology database is fully replicated on all the network nodes, and the structure of it is the same on the APPN nodes and the HPR nodes. Nodes in the base-APPN subnets see the HPR nodes and links as base-APPN nodes and links. However, nodes in the HPR subnet can distinguish between the base-APPN and the HPR nodes and links.

5.8.3.2 Directory Search

EN1 will send the Locate search request for LUy to its network node server NNA. If NNA has no previous knowledge of the location of LUy, and there is no central directory server in the network, then NNA will perform a broadcast search. The broadcast search will be sent to all the APPN and HPR network nodes in the network, using the CP-CP sessions in the usual way. NNE will send a reply indicating the location of LUy on EN2.

5.8.3.3 Route Computation

NNA, acting as the network node server for EN1, will calculate the route to be used through the whole network. The route calculation will be done in exactly the same way as in base APPN, using class of service and the topology database to select the least-weight route through the network. NNA has no knowledge that the HPR subnet is any different from the rest of the network. For the example, it is assumed that the route selected by NNA uses the following path:

EN1 - 1 - NNA - 2 - NNB - 3 - NNC - 4 - NND - 5 - NNE - 6 - EN2

The BIND from EN1 will be sent through the network with the RSCV, which was calculated by NNA. The RSCV is composed of a list of CP and TG vectors, as in base APPN.

5.8.3.4 BIND Routing through the HPR Subnet

Routing the BIND in the base-APPN subnets is always done using the RSCV. The BIND is sent to NNB using the RSCV calculated by NNA.

When the BIND reaches the APPN/HPR boundary function in NNB, an RTP connection will be set up, which crosses the HPR subnet and finishes in NND. If an RTP connection already exists between nodes NNB and NND for the requested class of service, this existing RTP connection will be used and a new RTP connection is not set up.

The routing of the BIND over the RTP connection through the HPR subnet is done using ANR rather than using the RSCV. After the BIND has left the APPN/HPR boundary function in NND, the RSCV routing continues for the last part of the route through the network.

The BIND response is sent on the reverse path. It uses the APPN connectors set up during the BIND request in the base-APPN subnets, and it uses the RTP connection set up across the HPR subnet.

HPR does not change the APPN route selection process. The RTP connection will follow the same route through the HPR subnet as was indicated in the RSCV calculated by NNA, and so will take the following path:

NNB - 3 - NNC - 4 - NND

However, if a failure occurs in NNC or either of the intermediate links 3 or 4 get disconnected, then a non-disruptive path switch will cause a different route for the RTP connection to be set up between NNB and NND. In this case, NNB or NND will calculate the new route for the RTP connection.

5.8.3.5 LU-LU Session Routing

The session traffic can then begin, and this will follow the path as described above. In the base-APPN subnets, the routing is done with intermediate session routing, using the session connectors set up during the BIND process. In the HPR subnet, the routing is done using ANR across the RTP connection.

As the APPN class of service is used to calculate the route through the HPR subnet, the transmission priority requested in the class of service is used in HPR when traffic is flowing through the network. HPR nodes establish queues for the four transmission priorities per out-going transmission group to prioritize traffic at the link level. In addition to this, the new ARB congestion control that is used

in the HPR subnet will regulate the HPR traffic and so the performance in the HPR subnet will be improved.

NNC, as an intermediate node on the path of the RTP connection, has no knowledge of the RTP connection, nor of the sessions that pass across it. It can route the packets of the LU-LU session traffic based on the ANR routing information that is contained in every packet.

5.8.4 Changes to the APPN Architecture

The two main components of HPR are the Rapid Transport Protocol (RTP) and Automatic Network Routing (ANR).

Apart from these two new components, HPR requires only minor changes to previously defined APPN functions. The main changes introduced by HPR are discussed in this chapter.

5.9 HPR Base and Options / Enhancements

The HPR function is an extension to the APPN architecture and can be added to an existing APPN end node or APPN network node.

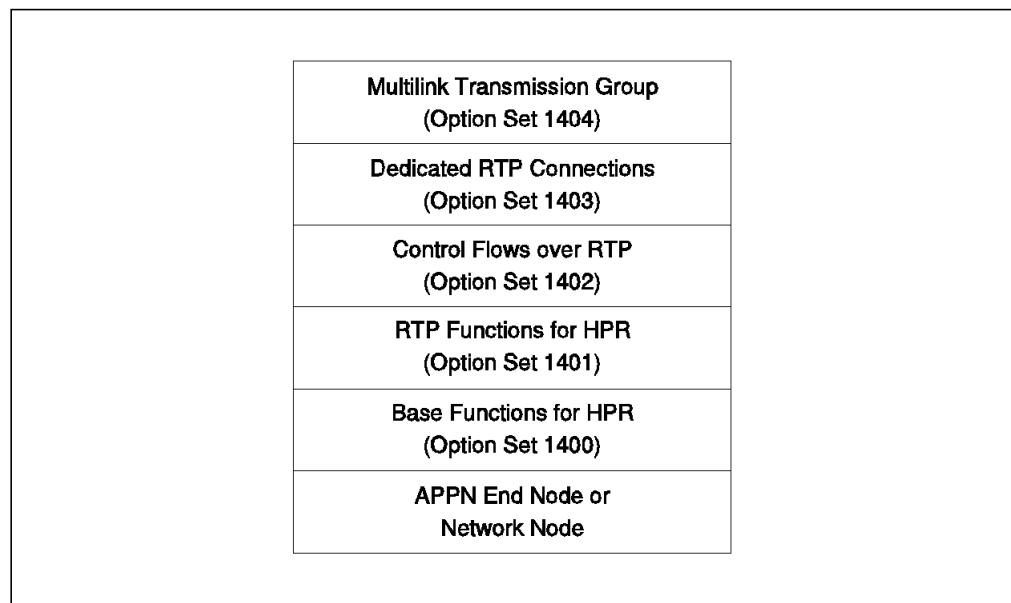


Figure 67. HPR Base and Options

5.9.1 Base Functions for HPR

The primary role of the HPR base functions (APPN option set 1400) is to support ANR. Products that implement only the HPR base functions will act as intermediate ANR routing nodes for RTP connections. Such nodes cannot act as endpoints for RTP connections. As APPN intermediate routing is done only by APPN network nodes, it follows that a node that is only supporting ANR must be an APPN network node. An APPN end node that implemented only the HPR base could not make use of the HPR ANR routing functions.

The following lists the HPR base functions:

Intermediate routing of network layer packets using ANR

HPR Network Layer Packets (NLPs) flowing on RTP connections and using ANR may be efficiently routed through the node.

Using FID2 routing for CP-CP sessions and route setup requests

All CP-CP session traffic between a base HPR node and its neighbors flows (as in base APPN) using FID2 PIUs.

Prior to establishing an RTP connection, a route setup protocol is executed in order to obtain the necessary ANR information associated with each link along the path. Every node along the path, including base HPR nodes, participates by adding the appropriate ANR information. When the route setup messages are exchanged between an HPR base node and its neighbors, they flow in FID2 PIUs.

FID2 PIUs and NLPs share a link

Both FID2 PIUs and NLPs may flow over a single link. They are distinguished by the first 3 bits in the packet (B'001' for a FID2 PIU and B'110' for an NLP).

Using FID2 routing for LU-LU sessions that use intermediate session routing (as opposed to ANR)

APPN LU-LU traffic not flowing over RTP connections continues to use FID2 PIUs.

HPR capability exchange via XID3

During XID3 exchange, an HPR node indicates its level of HPR support.

A maximum packet size of at least 768 bytes on an HPR link

Any link that supports HPR must be capable of transporting packets at least as big as 768 bytes.

TDUs indicate level of HPR support

TDUs for TGs and nodes are sent indicating the appropriate level of HPR support.

Calculation of HPR-only routes

HPR network nodes are able to calculate routes that contain only nodes and TGs that support HPR.

Link-level error recovery support

Link-level error recovery is always required for the following link types (not using link-level error recovery on these link types is not allowed or possible):

- IBM-compatible parallel and ESCON channels
- X.25
- SDLC

The ability to send packets over a link without using link-level error recovery is required support for all other link types (not listed above) supported by HPR. Using link-level error recovery on these links is optional.

5.9.2 RTP Functions for HPR

Rapid Transport Protocol (RTP) is the transport protocol used in HPR for transporting data across HPR subnets.

A node that supports the RTP functions for HPR (APPN option set 1401), in addition to the HPR base functions, can act as an endpoint of an RTP connection. RTP connections can only be established between nodes that support the RTP functions. To be able to make use of HPR in a network, there must be an HPR subnet with at least two nodes that support the RTP functions. If all the nodes in a network support only the HPR base functions, then no RTP connections can be established and the network will run base APPN protocols.

The endpoint of an RTP connection can be in an APPN end node or an APPN network node, and so the RTP functions for HPR can be implemented in either node type.

The following lists the RTP functions for HPR option set:

Rapid Transport Protocol (RTP)

This is the transport protocol used in HPR for transporting data across HPR subnets.

Non-disruptive path switch

If the current path used by an RTP connection fails, the connection may be automatically switched to a new path. Sessions that are transported over the RTP connections are not disrupted.

APPN/HPR boundary function

APPN (FID2 PIU traffic) is mapped to HPR (NLP) traffic and vice versa.

Directory reply with LU's network connection endpoint (NCE) identifier

An NCE identifier is an ANR label that allows an NLP to be routed to a specific component within a node. The component is uniquely identified by the label. A search reply for an LU contains the NCE identifier associated with that LU.

Note

eNetwork Communications Server for OS/2 Warp has not implemented routing to multiple components within the node. Therefore, there is only one NCE and it is always 0x80.

Of course, all the base functions for HPR are also supported by a node that implements the RTP functions for HPR.

5.9.3 Control Flows over RTP

A node that implements the HPR control flows over RTP option (APPN option set 1402), in addition to the RTP functions for HPR option set, will support the use of:

- RTP connections for CP-CP sessions
- RTP connections for route setup requests and responses

In each case, these RTP connections terminate in the adjacent nodes. Only when both HPR nodes, connected by one or more HPR links, support the control flows over RTP option, will RTP connections and network layer packets be used

to transport the CP-CP session flows and route setup requests and responses; otherwise, FID2 routing will be used.

Note that because FID2 routing is not supported over HPR multi-link transmission groups, both nodes connected by a multi-link transmission group have to support the control flows over RTP option.

5.9.4 A Sample HPR Network

Figure 68 shows a sample network that includes different possibilities of HPR base and options.

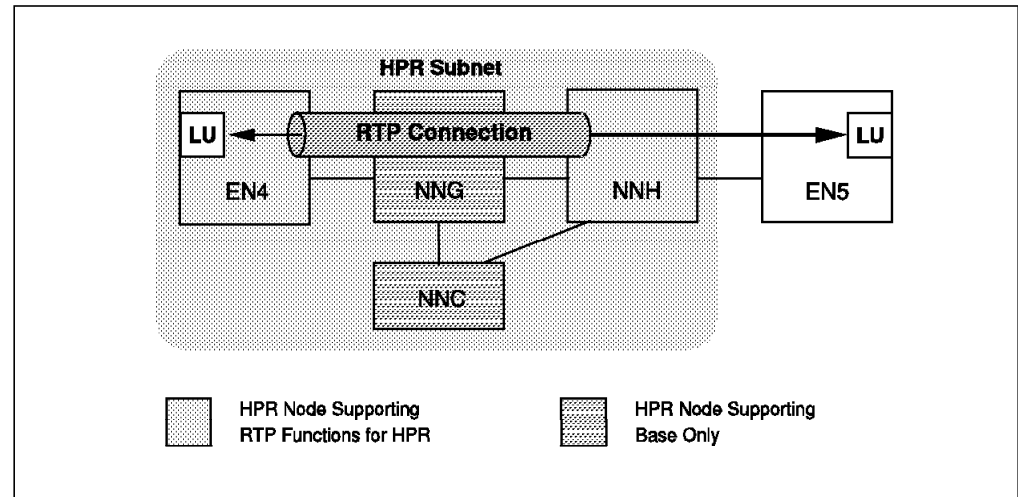


Figure 68. HPR Base and Options Example

In this example, it is assumed that the end nodes are attached to their network node servers via unreliable lines that require link-level error recovery. The network nodes are assumed to be connected via a backbone of high-speed, reliable lines that do not require link-level error recovery.

The link between EN4 and NNG is an HPR link, and to support the unreliable link, link-level error recovery is used on the link. The link between NNH and EN5 is an APPN link and so link-level error recovery is provided as part of base APPN.

EN4 and NNH provide endpoints for the RTP connection and so require the RTP functions for HPR. NNG supports ANR for the RTP connection and NNC may provide ANR after a non-disruptive path switch. Thus, NNG and NNC need the HPR base function, but because they are not RTP connection endpoints, they do not require the RTP functions for HPR.

Thus the HPR options are implemented in the nodes in the HPR subnet in Figure 68 as follows:

EN4	HPR base and RTP functions for HPR
NNG	HPR base
NNH	HPR base and RTP functions for HPR
NNC	HPR base

5.10 Adaptive Rate-Based Flow/Congestion Control

The adaptive rate-based (ARB) congestion and flow control algorithm is designed to let RTP connections make efficient use of network resources by providing a congestion avoidance and control mechanism.

The basic approach used in this algorithm, as the name implies, is to regulate the input traffic (offered load) of an RTP connection based on conditions in the network and at the partner RTP endpoint. When the algorithm detects that the network or the partner endpoint is approaching congestion and the path becomes saturated, resulting in increased delays and decreased throughput, it reduces the rate at which traffic on an RTP connection is allowed to enter the network until these indications go away. When the network or partner endpoint is sensed to have enough capacity to handle the offered load, the algorithm allows more traffic to enter the network without exceeding the rate that the slowest link on the path of an RTP connection or that the receiver can handle.

5.10.1 ARB Operating Region

Figure 69 on page 113 shows the network throughput as a function of the offered load for a given path.

The *knee* (point K) is the point beyond which the path starts to get saturated because transmission queues are developing, resulting in higher network delays. An increase in offered load (such as the sending rate) then does not correspond to an increase in throughput which is reflected in the receiving rate. ARB detects this saturation condition and adjusts (reduces) the sending rate accordingly, thus preventing operation beyond the *cliff* (point C). Because HPR does not use a hop-by-hop flow/congestion control algorithm (as base APPN does with adaptive session-level pacing), intermediate nodes will drop packets when their buffers are depleted. The cliff reflects the point beyond which there is a significant loss of packets because of excessive queueing along the path. An increase in offered load beyond this point results in a drastic decrease of throughput because of packet retransmissions.

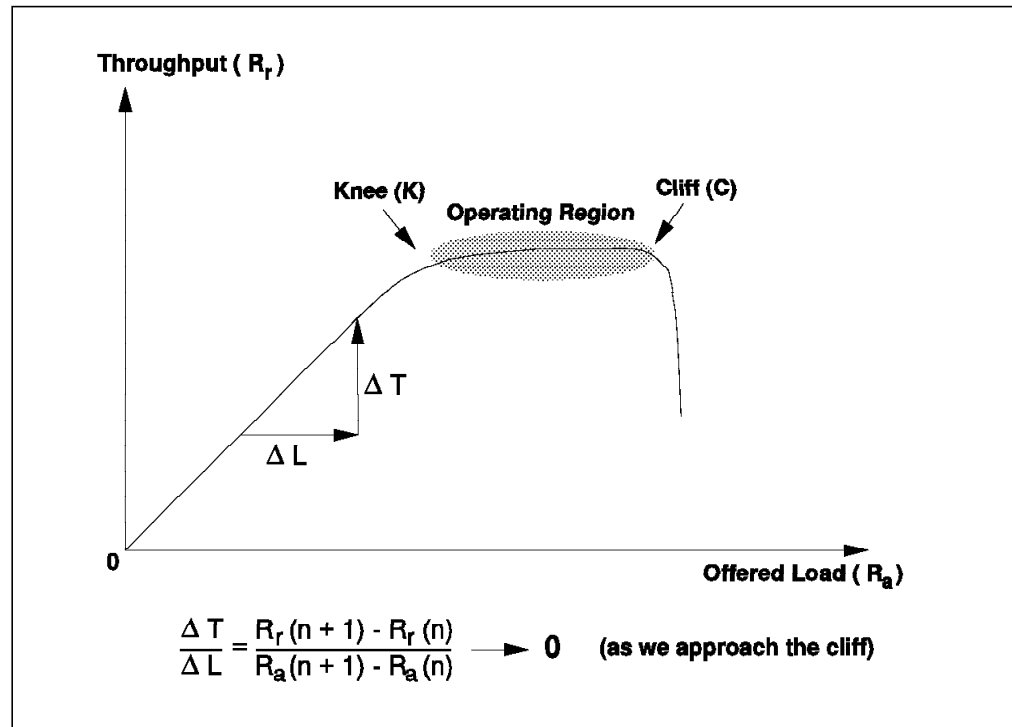


Figure 69. ARB Operating Region

5.10.2 ARB Principles

The ARB algorithm has the following properties:

- It is adaptive to network conditions in such a way as to maximize throughput and minimize congestion.
- It smooths the input traffic into the network by avoiding large bursts when the physical capacity of the access link is larger than the allowed sending rate. This prevents long queues from developing in the network and helps minimize oscillation in the network traffic patterns.
- It provides effective end-to-end flow control between RTP endpoints.
- It is simple to implement and requires minimum overhead, both in processor cycles and network bandwidth.
- It is generally fair in providing equal access to network resources between all RTP connections.

The ARB algorithm employs a closed-loop, distributed control mechanism based on information exchanged between the two endpoints of a connection. Figure 70 on page 114 shows the relationship between an ARB sender and an ARB receiver over an RTP connection between two RTP endpoints.

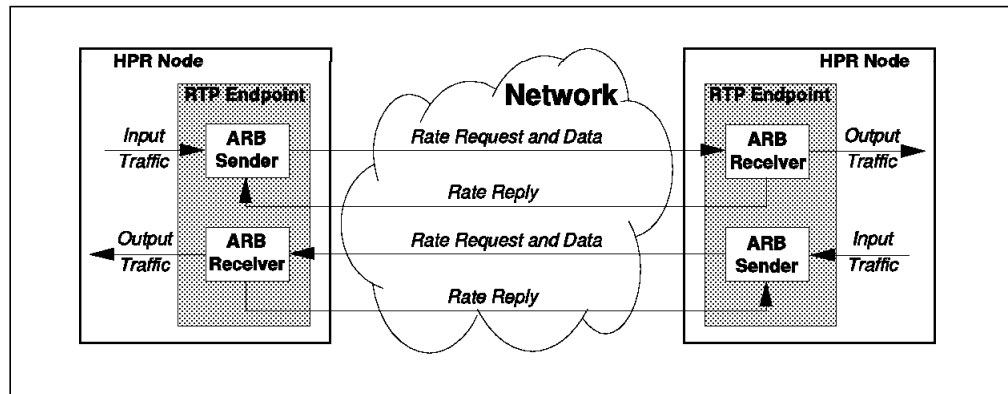


Figure 70. ARB Closed Loop

The ARB algorithm is implemented in the endpoints of an RTP connection. At each endpoint there are two components, an ARB sender and an ARB receiver. The ARB procedures performed by the sender and receiver at one end are the same as those performed by the sender and receiver at the other end respectively. (Note that intermediate nodes have no awareness of the ARB protocol and do not participate in it.)

The ARB algorithm always regulates the rate at which data is flowing from the ARB sender to the ARB receiver. The sender continually queries the receiver, by sending a *rate request* (along with user data) in order to obtain information about the state of the network and the state of the node containing the receiver. The receiver responds by sending back a *rate reply*. The sender then adjusts its send rate based on the information in the rate reply. The sender may reduce its send rate to relieve congestion or increase it to take advantage of the available network capacity.

Fixed characteristics of the path (that is, the speed of the slowest link along the path and the total transmission delay over the entire path) are factored into the ARB algorithm at both the sender and the receiver. These path characteristics are communicated in the ARB setup message during RTP connection establishment (including non-disruptive path switch). The ARB messages (rate request, rate reply, and setup) are transmitted in the ARB optional segment of the THDR and piggybacked (whenever possible) onto normal data packets.

5.10.3 ARB Algorithm

As is illustrated in Figure 71 on page 115, at regular intervals approximating the round-trip delay on the RTP connection, the ARB sender sends a rate request in an ARB segment, which is always added to a packet containing user data. (When no user data is flowing on an RTP connection, there is no need to measure the RTP connection's performance.) This rate request includes the *sender's measurement interval*, which is the time that has elapsed since the last request was sent. Upon receipt of the request, the receiver determines whether any delay has occurred in the network. It does so by calculating the difference between the sender's measurement interval and the *receiver's measurement interval*, which is the time that has elapsed since the last rate request was received.

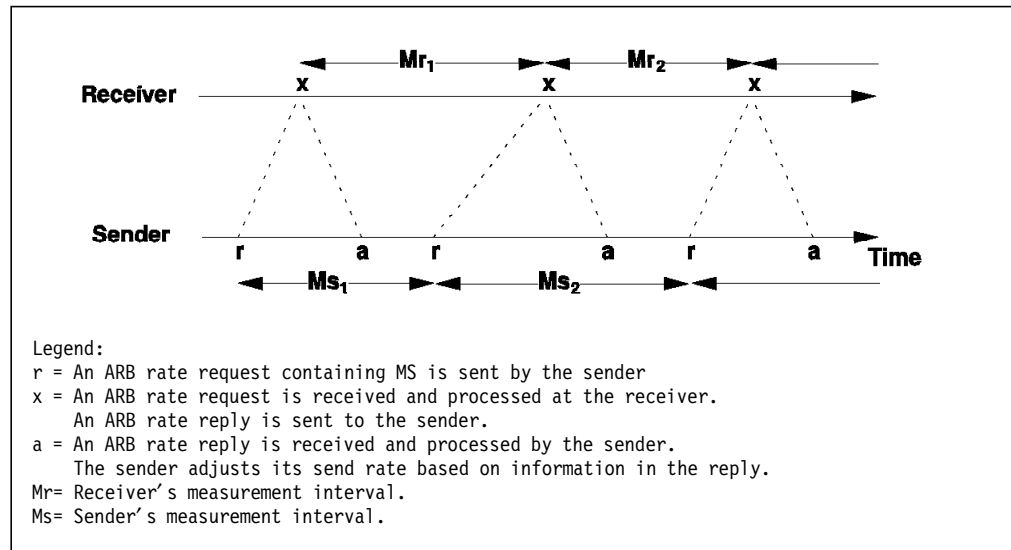


Figure 71. Overview of ARB Algorithm

The receiver also takes into account previous delays remembered from earlier rate request messages. Based on the measured changes in network delay, the receiver will then recommend appropriate actions to be taken by the sender. These recommended actions are communicated in a rate reply message that enables the sender to adjust its send rate appropriately. The ARB segment containing the rate reply may be either carried in the THDR of a packet with user data or, if none is available, sent alone as a packet without user data. The receiver, in addition to deriving its recommendations based on network delays, can also tell the sender to adjust its send rate based on conditions within the receiving node (for example, buffer depletion).

5.10.4 Non-Disruptive Path Switch

The HPR non-disruptive path switch function is used to automatically route RTP connections around failed links or nodes. This function operates only in an HPR subnet, and not within or across a base-APPN subnet. When a failure occurs and an HPR-only alternate path exists that satisfies the requested class of service (COS), the traffic of the RTP connection using the failed path is rerouted over the new alternate path in a manner that is transparent to the sessions being carried over the RTP connection.

If the original HPR path recovers before a path switch occurs, then the path will not be switched.

Figure 72 on page 116 shows a failure in a sample network on the link between NNC and NND.

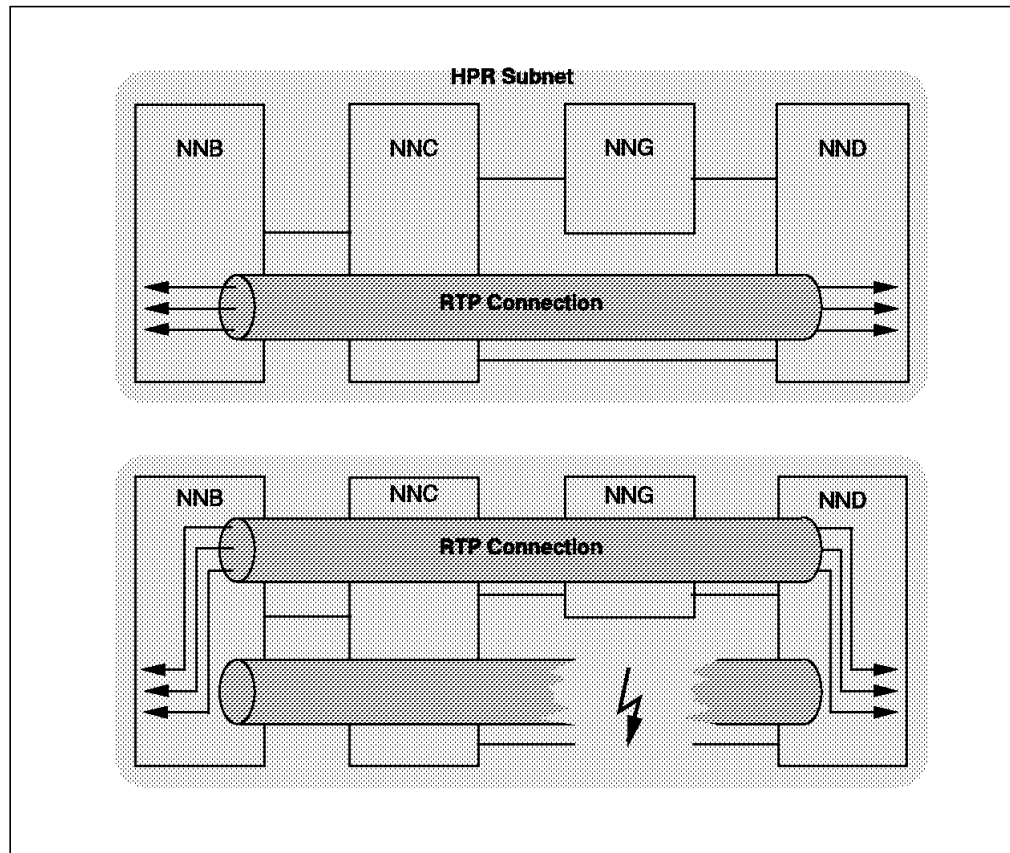


Figure 72. Non-Disruptive Path Switch in an HPR Subnet

Which RTP partner initiates the path switch depends on the partner types. *Mobile* partners prefer to initiate a path switch, whereas *stationary* partners yield to the partner's wishes. There are then three possible combinations:

- If both partners are stationary or both are mobile, either partner may initiate a path switch.
- If one partner is stationary and the other is mobile, then:
 - Only the mobile partner initiates the path switch when a connection failure is detected.
 - Either partner may initiate the path switch in the other cases described in the next section.

The RTP origin (the node that initiates the RTP connection) communicates its type (mobile or stationary) during RTP connection setup in the Switching Information (SI) segment. The RTP destination communicates its type in the route setup reply.

5.10.4.1 Path Switch Triggers

Various circumstances can trigger a non-disruptive path switch:

RTP connection failure detection

If a message is sent requesting an acknowledgment and the Short-Request timer expires before a reply is received, the sender attempts to determine whether the partner is still reachable. (See *Inside APPN - The Essential Guide to the Next-Generation*, SG24-3669 for a description of the Short-Request timer.) A state exchange message is sent including the status of the sender and asking for the

status of the receiver. If the Short-Request timer expires again for this message it is retried until the retry limit is reached. Finally the sender concludes that the connection has failed and triggers a path switch.

Support for this trigger is required in all nodes implementing the RTP functions for HPR.

Local link failure

If a local link being used by an RTP connection fails, it may trigger a path switch. This allows RTP connections to be switched faster than waiting for each RTP connection on a link to time out individually.

Support of this trigger is optional.

Remote link failure

If a TDU is received indicating that a remote link used by an RTP connection has failed, this may trigger a path switch. This can happen only when the local node is an NN (ENs don't receive TDUs) and for RTP connections that end in NNs (ENs don't send TDUs).

Support of this trigger is optional.

Operator request

The node operator or a network management operator requests that the path be switched. This request may be done for a specific path. This function is especially useful for switching an RTP connection back to its original path after a non-disruptive path switch has occurred and once the original path is operational again.

Support of this trigger is optional.

5.10.4.2 Path Switch Timer

Once it is determined that a path switch needs to be done, a path switch timer is started. This timer indicates the time allowed to accomplish the switch. If this timer expires and the path has not been successfully switched, the RTP connection is deemed to have failed.

The path switch timer time-out value is usually associated with the transmission priority. Suggested timer default values for each priority are:

- 1 minute for network priority
- 2 minutes for high priority
- 4 minutes for medium priority
- 8 minutes for low priority

Note that the use of a path switch timer handles the case where a path switch is attempted before the TDU, indicating the link failure that caused the path loss has arrived in all nodes. In this case, the same (bad) route might be calculated again and the RTP retries will fail again. This procedure could be repeated several times before the TDU arrives and a good path can be calculated. For this reason, products may find it desirable to wait between path switch attempts.

Non-disruptive path switch can be disabled by setting the path switch timer to a value of zero. In this case, RTP will not attempt a path switch when the RTP connection fails.

5.10.4.3 Obtaining a New Path

When an attempt is made to obtain a new path for an RTP connection during a path switch, the new path has to be an *HPR-only* path (that is, a path between the two RTP endpoints that contains only HPR-capable links supporting ANR routing). New paths are represented by RSCVs (just as in base APPN).

Obtaining a new path may involve some or all of the following functions:

Directory search

The target resource used for directory searches is always the CP name of the remote RTP partner. A new indicator on the directory search request specifies that an HPR-only path is requested.

RSCV calculation

All HPR NN servers (NNs that support the HPR base function) can calculate HPR-only paths. Route Selection Services will select the *lowest-weight HPR-only route*. If there are two possible routes for the required class of service, one that passes into a base-APPN subnet and one that only uses the HPR subnet, then the HPR subnet route will be selected, even if the weight is higher.

Note that if an HPR EN is connected to both an HPR NN and to a base-APPN NN, but the base-APPN NN happens to be its NN server, the request to obtain an HPR-only path might fail because the base-APPN NN does not understand the request to calculate an HPR-only path. The EN then has to check whether the path is indeed HPR-only and if it is not then the RTP connection fails. HPR-only paths can be recognized by examining the returned RSCV to see if all the links in the RSCV are HPR capable.

Route setup protocol

The route setup must always be performed to obtain information for the new path. See Figure 73 on page 119 for detailed information about the route setup command contents.

```

Line: 6838 Send MU
Time stamp: 16:45:49.45
DLC type: HPR
TCID: 000000CF
Transmission priority: Network
TH: FID2, Exp, OIS, LFSID=0x00000, SNF=0x0000
RH: RQ, NC, FI, OIC, RQN
Route setup
  Type = Request
  Route Setup triggered by path switch = No
  Destination hop index = 1
  Locate search is required = No
  Destination is mobile = No
  NCE is used for all LUs/BFs = No
  Path switch time (milliseconds) = 0
  Network name control vector:
    (0x0002) Network name type = LU
    (0x0003) Name = USIBMRA.SERVERA
  Route selection control vector:
    (0x0002) Maximum hop count = 1
    (0x0003) Current hop count = 1
  TG descriptor control vector
    TG Identifier TG Descriptor subfield
      (0x0002) TG number = 10
      (0x0004) Partner name = USIBMRA.SERVERA
      (0x0013) Link connection network = No
      Additional configuration information = No
      HPR = Supported
      TG type = Boundary Function based or APPN
      Intersubnet link = No
      Extended border node = Not supported
      RTP Tower = Supported
    TG Identifier Extension
      (0x0002) Branch Extender uplink = Yes
  Fully qualified PCID control vector:
    (0x0002) PCID = 0xc0c3954535943e99
    (0x000B) Network qualified CP name = USIBMRA.SERVERB
  COS/TPF control vector:
    (0x0002) Transmission priority = High
    (0x0004) COS name = #INTER
  Route Information control vector
    (0x0002) Route direction = Forward
    REFIFOing required = Yes
    (0x0004) Maximum packet size = 1033
    (0x0008) Accumulated transmission time (microseconds) = 63157
    (0x000C) Minimum link capacity (Kbits/second) = 19
    (0x0010) Limited resource liveness timer (seconds) = 0
  ANR Path control vector
    (0x0003) ANR label represents a subarea network route = No
    (0x0004) ANR label = 8078

```

Figure 73. HPR Route Setup Command

5.11 Configuration Services

Configuration Services in HPR nodes basically works the same as in base APPN. There are only a few additions necessary to support HPR specifics.

5.11.1 HPR Data Link Control

HPR is an enhancement to APPN and can operate over links supported by base APPN. Therefore, hardware adapters and DLCs currently being used for APPN can be used for HPR. The actual selection of DLCs supported for HPR is a product implementation decision.

5.11.1.1 Maximum Packet Size

The maximum packet size supported for any HPR link frame must be at least 768 bytes. The NHDR and THDR cannot be segmented; therefore the supported maximum packet size on any link must accommodate the largest possible (within reason) NHDR/THDR combination. The size of the NHDR depends on the number of hops (TGs) and the size of the ANR labels used. The largest possible THDR is the one used when activating an RTP connection. A maximum packet size of 512 bytes would satisfy the above in almost all cases, but would not leave much space for future expansion. The maximum packet size of at least 768 bytes was therefore chosen to ensure that route setup requests and replies are never segmented.

Because any given network layer packet has to be able to be sent over any link in an HPR multi-link transmission group, the maximum packet size must be the same for all links within an MLTG.

5.11.1.2 HPR Link Formats

Some changes are required in the link header to support traffic with no link-level error recovery. This is discussed in more detail in 5.11.6, "Link Data Traffic" on page 128. The link trailer is unchanged and contains the frame check sequence field that is the result of cyclic redundancy checking (CRC). The CRC applies to both the link header and the packet. On HPR links that are not using link-level error recovery, the CRC is the only integrity check and is therefore always required in HPR.

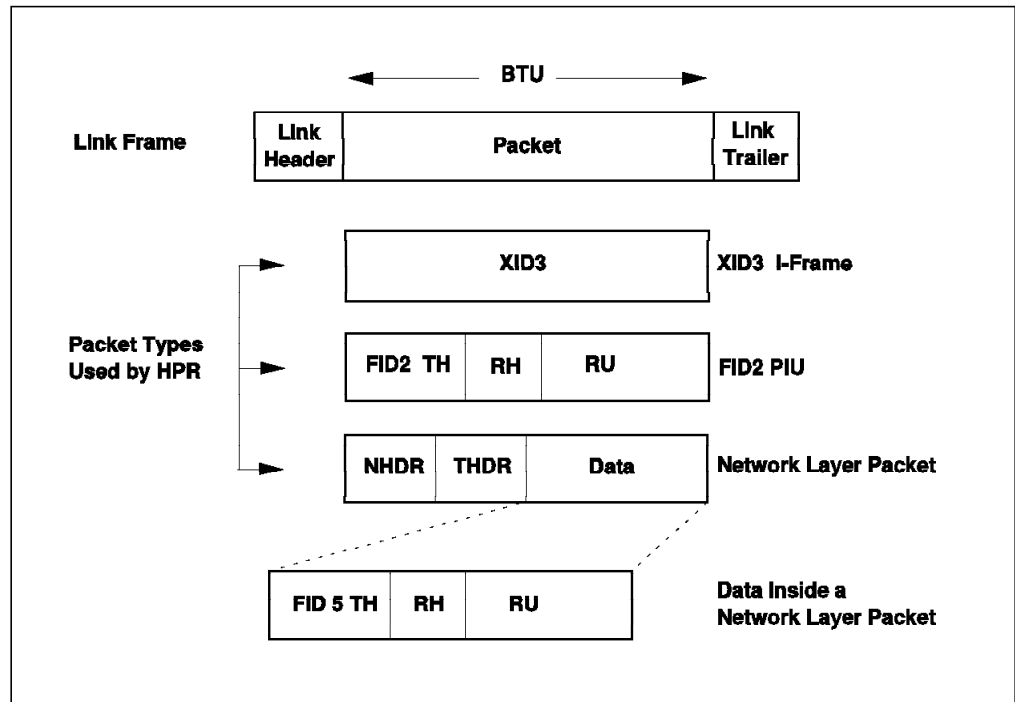


Figure 74. HPR Packet Formats

Figure 74 shows the different packet types that are used by HPR. Each of these is discussed below:

XID3 I-Frame

The XID3 format is similar to the existing format, with the addition of a new HPR Capabilities control vector.

FID2 PIU

FID2 PIUs are still supported in HPR and the structure of them is exactly the same as in base APPN. HPR uses FID2 PIUs in the following circumstances:

- LU-LU sessions transported using intermediate session routing (not HPR)
- CP-CP sessions between nodes that do not both support the HPR control flows over RTP option
- Route setup requests between nodes that do not both support the HPR control flows over RTP option

Network Layer Packet

The new network layer packet is used by HPR when doing ANR for an RTP connection. Two new headers are used in HPR for ANR and to control the RTP connection:

NHDR

This is the network layer header that contains the ANR routing information.

THDR

This is the RTP transport header that contains the RTP transport information.

The data inside a network layer packet can itself be of two types:

- LU-LU or CP-CP session traffic. This session traffic will use a new FID5 transmission header in place of the base APPN FID2 header.

- Route setup messages in the form of GDS variables.

An eNetwork Communications Server for OS/2 Warp node supports ANR and FID2 routing on the same HPR link. This support is part of the base HPR function.

```

Line: 9802 Send RTP Header
Time stamp: 16:45:50.79
DLC type: HPR
TCID: 0x000000D0
| RTP Header
|   Switching mode = ANR
|   Transmission priority = High
|   Packet is time sensitive = Yes
|   Minor congestion exists = No
|   Significant congestion exists = No
|   ANR Routing = 0x807880ff
|   TCID assigned by sender = No
|   TCID = 0x0000000000000048
|   Connection Setup segment present = No
|   Start of message = No
|   End of message = No
|   Receiver must reply with a status segment = No
|   Receiver must reply ASAP = No
|   Sender will retransmit this packet = Yes
|   Last message on this connection = No
|   CQF type = Not present
|   Optional segments present = Yes
|   Data offset = 0x0034
|   Data length = 0x00000000
|   Byte sequence number = 0x00000197
|   Optional segments:
|     Status segment
|       Packets have been lost = No
|       Connection is idle = No
|       Status report number = 0x0001
|       Status acknowledgment number = 0x0001
|       Received sequence number = 0x000000da
|       Delivered sequence number = 0x00000000
|     Adaptive Rate-Based segment
|       Message type = Rate reply
|       Rate adjustment action = Normal

```

Figure 75. HPR RTP Header

5.11.2 Limited Resource

HPR nodes support limited-resource links (see *Inside APPN - The Essential Guide to the Next-Generation*, SG24-3669 for a description of the existing APPN support). HPR nodes that contain the RTP connection endpoints have session awareness and can deactivate limited-resource links based on session usage.

HPR network nodes can also act as intermediate nodes doing ANR routing. HPR intermediate nodes have no session awareness and so cannot use the existing support to deactivate limited-resource links. So, HPR nodes will deactivate a limited-resource link when *both* the following are true:

- No known sessions are using the link (as FID2 sessions may still be supported over an HPR link).
- No traffic has used the link for a certain period of time.

5.11.3 HPR Connection Network Support

An RTP connection that supports LU-LU sessions can pass over a connection network. In base APPN, the TG that describes the link across the connection network to the real partner node is set up at session activation time. If an HPR route setup needs to pass over a connection network, the TG to the real partner node is required. This means that the TG needs to be activated earlier in the session establishment process for HPR.

The TG across the connection network is activated at route setup time. After the TG is activated, and if both nodes connected by the TG support the HPR control flows over RTP option, a long-lived RTP connection is established between the real nodes. If at least one node does not support the HPR control flows over RTP option, then FID2 packets are used to forward the route setup request. In either case, the route setup request is sent to the real partner node, and the ANR labels of the link are added at this time. For more details, see *Inside APPN - The Essential Guide to the Next-Generation, SG24-3669*.

Note that the dial information to establish the direct link to the real partner node is used only when activating the TG, just as in base APPN. This dial information is obtained from the RSCV as in base APPN, only now it is carried in the route setup request. Once the link is activated the route setup RTP connection can be established across the link (if both adjacent nodes support the control flows over RTP option) and the route setup request is forwarded over the link.

5.11.4 Multi-Link Transmission Groups

A multi-link transmission group (MLTG) consists of multiple DLC-level connections between two nodes made to appear to higher layers as a single connection. An MLTG is available for service as long as one or more of its constituent links are available. See 5.4, "Transmission Groups" on page 93 for explanations of the transmission group terminology.

Multi-link transmission groups are supported in traditional subarea SNA networks and in APPN HPR networks, but not in base APPN.

Although superficially similar to multi-link transmission groups in subarea networks, MLTGs in APPN HPR networks are significantly different in operation. This section describes HPR MLTGs.

5.11.4.1 HPR MLTG Requirements

MLTGs have advantages over single-link TGs and parallel TGs in a number of cases:

Where the traffic demand can exceed existing TG capacity

Traffic demand can exceed existing TG capacity when a single session reaches the point at which it needs more bandwidth than the TG can provide. Aggregate available bandwidth can be raised simply by the dynamic addition of more links. If the demand subsequently falls, the extra bandwidth can be taken back by deletion of the extra links, saving network charges. Parallel TGs cannot help in this circumstance.

The need may also arise because of varying loads placed on a TG by a collection of sessions, rather than any single session. In this instance, adding parallel TGs *might* be an alternative solution, or not, depending on class-of-service and route selection implementations. But a single session could not use more capacity than the link offers that carries this session.

Where multiple lower-speed links are less expensive than a single higher-speed link

There are cases where multi-link transmission groups prove less expensive than single-link TGs. In certain countries circuit capacities of 64 kbps and 2 Mbps are available, but nothing in between. If you live in one of these countries and have to provide 100 kbps of bandwidth, for example, you may find it costs less to put two 64 kbps links into a multi-link transmission group than to have a single 2 Mbps link.

Where individual links are unreliable

Although HPR provides a fast non-disruptive path switch capability, not even this will be necessary if your TGs never fail. If you are considering MLTGs to avoid TG failures however, you must plan for the potential effects of temporarily reduced TG capacity. When one of several active links in an MLTG fails, effective capacity will be reduced even though the TG does not itself fail.

Where you have a subarea network including multi-link transmission groups

If you have grown used to having the multi-link transmission group facility in subarea networks you may feel more comfortable about migration to APPN HPR, knowing a similar facility is there.

Additional design objectives of the MLTG architecture include:

- The need to support mixed link types within MLTGs
All supported SNA link types are also supported in HPR MLTGs.
- The need to support mixed link speeds within MLTGs
- The need to minimize system definition

5.11.4.2 HPR MLTG Overview

The critical parameter determining whether two links belong to one MLTG or to two parallel TGs is TG number (given of course that the links connect the same pair of nodes). If the links share the same TG number, then they belong to an MLTG; if they have different TG numbers, then they belong to parallel TGs. In this regard, subarea SNA and HPR do not differ.

One of the architectural problems with subarea multi-link transmission groups was the need for resequencing of packets. Higher layers required DLC to guarantee delivery of packets, hop-by-hop, and to guarantee FIFO order. This dictated, among other things, that SNA subarea nodes had to act as *store-and-forward* switches, being unable to make forward routing decisions until entire packets had been safely received. It could easily happen that two packets, transmitted on different links within a multi-link transmission group, would reach this point in reverse order of their initial order. The receiving node would have to buffer the second packet, pending the arrival of the first. This TG resequencing function could impose large processing overheads, especially where there were widely varying line speeds, propagation delays, or packet

lengths, or where there were significant line error rates. In today's high-speed networks, resequencing delays enroute would be unacceptable.

HPR eliminates the need for TG resequencing and for hop-by-hop error recovery by shifting these functions to RTP endpoints. When a VR-based transmission group (VR-TG) crossing the subarea network includes a subarea multi-link transmission group, resequencing is not done for HPR network layer packets transported over that subarea MLTG.

In the HPR MLTG architecture, error recovery on individual links is optional, and TG resequencing enroute is absent. Because FID2 packets have to be transmitted reliably and in sequence, HPR MLTGs do not support any FID2 traffic. HPR MLTGs must carry ANR network layer packets exclusively. This means, in turn, that RTP connections must be used for CP-CP sessions and route setup flows. Both nodes connected by an HPR MLTG must support the control flows over RTP option.

As regards routing and ANR labels, MLTGs are treated the same as single-link TGs. See *Inside APPN - The Essential Guide to the Next-Generation*, SG24-3669. An MLTG is assigned one ANR label for each direction.

MLTGs and single-link TGs are also considered alike by TRS when it comes to the generalities of topology databases, TDUs, and route calculations. Differences show up when an MLTG's characteristics change *in flight*, for instance, when a new link is added. Such circumstances cannot arise in single-link TGs. When MLTG characteristics do change, topology database records are modified and TDUs generated. See *Inside APPN - The Essential Guide to the Next-Generation*, SG24-3669.

Some functions are not supported in HPR MLTG:

- Limited resource
- Connection networks
- Non-activation XID

Much of the HPR MLTG architecture revolves around the handling of TG numbers and other characteristics governed by XID3 exchanges during link activation. In particular, it deals with the exceptions that can occur when differently defined links are put together.

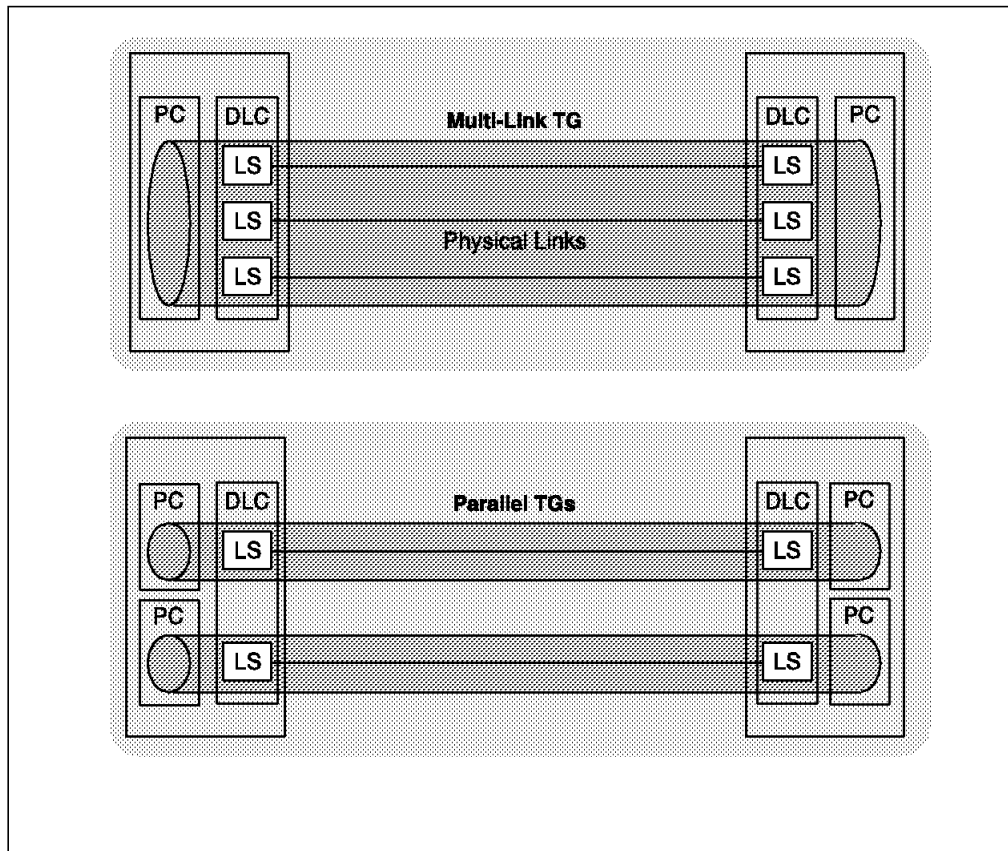


Figure 76. Multilink and Parallel TGs

5.11.5 HPR Link Activation

At link activation time, DLCs are started by HPR in the same way as in base APPN. XID3s are exchanged and the appropriate set mode signals are sent when the exchange is complete. See *Inside APPN - The Essential Guide to the Next-Generation*, SG24-3669 for a description of the existing APPN support.

The HPR DLC adds a new *HPR Capabilities control vector* to the XID3 used during the negotiation-proceeding phase. This control vector indicates the following:

- Whether link-level error recovery is required on this link
- Whether the RTP functions for HPR are supported by the node
- Whether the HPR control flows over RTP option is supported by the node
- Whether MLTG is supported and it is desired that this link become part of the specified MLTG (as indicated by the XID TG number)

The presence of the new control vector indicates base HPR support. If both nodes send the control vector in their XID3, the link is known as an *HPR link*.

If one node is an HPR node and the other is a base APPN node, then this link is an APPN link and the HPR protocols are not used. It is also possible (although not generally recommended) for an HPR node to activate a link in the APPN way, without including the new HPR control vector in the negotiation-proceeding XID3. This might be desirable in some environments to run very slow-speed links the APPN way because of constraints on link buffer sizes and bandwidth.

CP-CP session activation is triggered by link activation in the same manner as in base APPN.

5.11.5.1 MLTG Negotiation

An MLTG may be one of up to 20 *explicit* MLTGs connecting a pair of nodes, or it may be the only *default* MLTG. The difference between them lies simply in the TG number used: explicit MLTGs use numbers 1-20 inclusive, while the default MLTG uses TG number 240, formerly reserved. The point of having a *default* MLTG is explained in 5.11.5.2, “Advantages of the Default MLTG.”

The nodes agree about which TG number to use in essentially the same way as base APPN nodes agree about single-link TGs, that is by exchanging XIDs. This is described in *Inside APPN - The Essential Guide to the Next-Generation, SG24-3669*. One modification is needed: an MLTG-supported indicator is now carried in the HPR Capabilities control vector in XID3.

The nodes use the TG number fields and the MLTG-supported bit in the negotiation-proceeding XID3 exchanges to determine whether a link being activated belongs to an MLTG, and, if so, to which one. The following tabulates the various possibilities:

Table 1. MLTG Negotiation		
Side X	Side Y	Result
M, 0	M, 0	Add link to default MLTG (240)
M, 0	M, 1-20	Add link to explicit MLTG identified by Side Y
M, 1-20	M, 1-20	Add link to MLTG identified by both sides if the numbers match; otherwise reject activation
M, 0	¬ M, any	Activate link as a single-link TG, negotiating TG number as usual
M, 1-20	¬ M, any	Activate link as a single-link TG if the numbers match; otherwise reject activation

Key:

M = MLTG support indicator ON

¬ M = MLTG support indicator OFF

5.11.5.2 Advantages of the Default MLTG

The idea of the default MLTG is to enhance operational flexibility. It can also reduce system definition effort. If you have the nodes at both ends of an MLTG default, you will eliminate TG number definitions altogether. If, on the other hand, you have a need for parallel MLTGs where you must, of course, use more than one TG number, you can have one side default and define the TG numbers on the other.

The default MLTG will be particularly useful in client/server environments where you might otherwise spend a lot of time administering TG numbers for your many client/server connections.

5.11.6 Link Data Traffic

The DLC formats used for network layer packets (NLPs) depend on the type of data link control and whether link-level error recovery is used on a link or not. (Note that FID2 PIUs are always sent using link-level error recovery, just as in base APPN.) Whether link-level error recovery procedures for NLPs for a link shall be used is determined during the XID3 exchange. HPR will change the existing DLC format used by base APPN only if no link-level error recovery is being done on the link.

At present, the HPR architecture describes the following DLCs:

Frame relay

If no link-level error recovery is used on a frame relay link, then base support is to carry NLPs in frames with the level 2 protocol identifier indicating that no IEEE 802.2 header is present. The HPR NLP then immediately follows the level 3 protocol identifier for HPR.

Multiple HPR links can optionally be multiplexed in a single frame relay virtual circuit by using different SAP fields in the 802.2 header. This support may be chosen to simplify the interface to a frame relay network and to reduce WAN connection costs (when using a frame relay carrier). In this case, NLPs are carried in frames with the level 2 protocol identifier indicating the presence of an 802.2 header. The 802.2 header then contains the SAP associated with the individual link and the UI command code (if not using link-level error recovery).

LANs

For performance reasons a separate SAP, different from the SAP currently being used for APPN traffic, is normally used to transmit NLPs with no link-level error recovery. (The default is X' C8' .) The SAPs to be used for NLPs with no link-level error recovery are exchanged between the two nodes activating a LAN link during the XID negotiation proceeding phase. NLPs requiring link-level error recovery use the same SAP as existing APPN traffic. If the HPR SAP is configured to have the same value as the APPN SAP, then NLPs with no link-level error recovery are sent as UI frames. Even though there might be two different SAPs to separate HPR and base-APPN traffic, there is logically only one link station. All traffic travels along the same physical path. Because there is only a single link station, all LLC commands and responses (XID, SABME, DISC, etc.) flow using the APPN SAP.

SDLC

There are no changes required to run HPR over SDLC. It might be possible to transmit NLPs over SDLC links without using link-level error recovery by sending them as UI frames, but the benefit (especially on multipoint connections) appears to be minimal. Since SDLC is not considered one of the high-speed link protocols of the future, no HPR enhancements are being made to it.

X.25

eNetwork Communications Server for OS/2 Warp Version 5.0 does not support HPR over X.25 networks.

5.11.7 After Link Activation

If both nodes support the HPR control flows over RTP option, then an RTP connection is set up between the adjacent HPR nodes over every link that is activated. This RTP connection is used during route setup, to carry the route setup requests and it remains active for as long as the link remains active (hence the term *long-lived RTP connection*). The route setup requests and the route setup RTP connection are explained in *Inside APPN - The Essential Guide to the Next-Generation*, SG24-3669. A link's long-lived RTP connection is established when the first route setup request arrives that has to be forwarded over this link.

CP-CP sessions are also activated after link activation. If both the adjacent nodes support the HPR control flows over RTP option, then an RTP connection is set up to carry the CP-CP sessions. If the nodes do not both support the HPR control flows over RTP option, then the CP-CP sessions will use FID2 protocols as in base APPN.

5.11.8 Link Failure Detection

When link-level error recovery is implemented, failure of a link is immediately detected. If a packet sent is not acknowledged, it is retransmitted for a defined number of times. If then no acknowledgment is received the link is inactivated and its changed status reported to topology and routing services. But when not using link-level error recovery for any packets sent over a link, this mechanism will not work simply because packets do not request an acknowledgment.

Since it is still necessary to detect link outages, a *link inactivity timer* is used. When no packets have been received for a certain time, an inactivity message is sent over the link requesting an acknowledgment. If this inactivity message then is not acknowledged within a given time (and after a number of retries), the link is inactivated and its changed status is reported to topology and routing services. The overall time to detect a link outage must be shorter than the end-to-end RTP connection timeouts in order for non-disruptive path switch to work properly. When a new path for an RTP connection affected by the link outage is calculated, the information about the link's status change must have been distributed in TDUs to all network nodes in the network.

There are actually three parameters that govern how long it will take to detect a link failure:

- When no packets have been received for the interval set by the *inactivity timer*, an inactivity message is sent to check if the link is still alive.
- The *send timer* defines how long the sender will wait for an acknowledgment of the inactivity message sent.
- The *number of retries* parameter defines how often the inactivity message is resent when it is not acknowledged within the send timer interval.

5.12 HPR and BrNNs

In contrast to the RSCV concerns associated with ISR, the source routing information in an HPR network layer header cannot confuse other HPR nodes, since ANR labels are variable length and only their owner interprets them. However, ensuring that the RSCV (which is used to get the labels) contains the right information is a little tricky.

In the event of link failure, HPR's nondisruptive rerouting can maintain sessions if an alternate route exists, but it cannot prevent session failures if the sole link configured between a BrNN and the WAN fails. In the event of a BrNN failure, HPR rerouting can maintain sessions if an alternate BrNN exists. If the traffic and tariffs warrant only a single peripheral link, this link could be backed up by a dial link that is activated only when the primary link fails. This is a common configuration requested by customers. If traffic and tariffs warrant concurrent multiple links between the branch and the WAN, they can back each other up.

With or without non-disruptive rerouting, HPR still provides significant benefits in a BrNN. ANR greatly reduces storage requirements in a BrNN when domain ENs support RTP. When the domain EN does not support HPR, a BrNN can still multiplex many sessions with endpoints in the BrNN's domain with the same COS on a single RTP pipe to the BrNN. HPR may significantly improve throughput, response time, WAN utilization, and storage.

A key aspect of the Branch Extender design for HPR is its use of true routes. Much HPR logic keys off data in the RSCV, rather than data from the directory search. By using true routes, many other details fall into place naturally.

The NCE scope indicator refers to the node whose CP name is in the TG vector in the RSCV, that is, the BrNN-served EN. Thus NCE Scope can be set to a value appropriate for the BrNN-served EN. This means endpoints in BrNN-served domains will cause no more directory searches than other HPR nodes.

Both RTP endpoints will have the same knowledge of the route, that is, the same hop count and the same hops. This means RTP connection reuse will occur as often with BrNN-served endpoints as otherwise.

5.12.1 RTP Connections Terminating on a BrNN

HPR needs to know if path switch requires a directory search to obtain tail vectors. This is the case when the RTP endpoint is on an EN. To review base HPR architecture, the route setup destination indicates it is an EN by setting byte 6 bit 0 in the Route Setup Reply GDS variable. The other partner indicates it is an EN by setting byte 2 bit 2 in CV83 on the RTP Connection Setup Request.

Before Branch Extender architecture, a node was either an EN or a NN. HPR learned this from the node definition. With Branch Extender architecture, a node presents an EN appearance on links defined as uplinks, and a NN appearance on links defined as downlinks. Therefore, the HPR component in a BrNN needs to know whether a link is an uplink or a downlink, so it can set these indicators correctly.

5.13 HPR Configuration

eNetwork Communications Server for OS/2 Warp allows you to configure HPR using the CMSETUP configuration utility program. However, there are some parameters that are only available in the NDF configuration file as explained later in this chapter.

You should always have in mind that if you enable a specific link or connection for HPR, it does not necessarily mean that HPR will be used. It actually means that the HPR options will be negotiated during the XID3 exchange when the link is established. Whether HPR is used or not will depend on the HPR capabilities

of the adjacent node to which the connection is made to (see Figure 77 on page 131).

```
Line: 1006 Send XID
Time stamp: 16:45:11.37
DLC type: SDLC
Adapter number: 00
ALS ID: 4419AF92BD9848AA
XID:
  (0x0000) Format = 3
           Node type = 2
  (0x0001) Total length = 131
  (0x0002) Node ID = 0x05d00002
  (0x0008) Init-self = Cannot receive
           Stand-alone BIND = Can receive
           .....
  (0x000F) Parallel TGs = Supported
           DLUR XID sender prefers ACTPU over CP-SVR pipe = No
           DLUS-served LU registration = Not supported
           Downstream Branch Extender = Not supported
           Upstream Branch Extender = Supported
  (0x0010) TG number = 10
  (0x0011) DLC type = SDLC
  (0x0012) DLC data length = 11
           .....
  (0x0064) HPR Capabilities control vector
           (0x0002) Error recovery = Required
                   RTP Tower = Supported
                   Control Flows over RTP Tower = Supported
           (0x0005) ANR label = 8078
  (0x006B) Token Ring/Ethernet 802.2 LLC subfield
           (0x0002) LLC SAP = c8
  (0x006E) Control Flows over RTP Tower subfield
           (0x0002) Maximum send packet size = 0x0409
           (0x0004) Path switch time = 60000
           (0x0008) Sender is mobile = No
                   Multilink transmission groups = Supported
           (0x0009) Control Point NCE instance identifier = 0x344bb549
           (0x000D) Route Setup NCE instance identifier = 0x344bb549
           (0x0012) Control Point NCE identifier = 80
           (0x0014) Route Setup NCE identifier = 80
```

Figure 77. HPR Supported Options in XID3 Exchange

5.13.1 LAN and Frame Relay - HPR Configuration

For implicit links, HPR support and options are configured in the data link control (DLC) profile as shown in Figure 78 on page 132.

For explicit links, HPR support and options can also be configured in the DLC profile but you can always override the HPR parameters in the connection configuration (define logical link) as shown in Figure 79 on page 133.

In addition, there are some HPR default parameters that you can only change in the node definition file (NDF) configuration as shown in Figure 82 on page 137.

5.13.1.1 DLC HPR Configuration

Select the HPR parameters option in the DLC profile to enable HPR, multilink transmission groups (MLTGs) if required and select HPR link level error recovery protocol (ERP) options to specify how links for this adapter will handle ERP when sending and receiving HPR packets.

If link level ERP is selected, HPR packets will be sent as numbered information frames (I-FRAMES). In this mode, the LLC2 (802.2) protocol is responsible for the detection and retransmission of lost frames.

If link level ERP is not selected, HPR packets will be sent as unnumbered information frames (UI-FRAMES). In this mode, the LLC2 (802.2) protocol cannot detect the loss of frames, so HPR's rapid transport protocol (RTP) must detect and recover lost packets.

Note: RTP is done at the connection's endpoints, while link level ERP is only done between two adjacent nodes within the connection.

The link level ERP selection will not disable RTP.

Select one of the following options:

- Link level ERP is not allowed (links for this adapter will not perform ERP).
- No link level ERP is preferred (links for this adapter will not perform ERP unless it is required by the partner). This is the default.
- Link level ERP is required (links for this adapter will perform ERP)

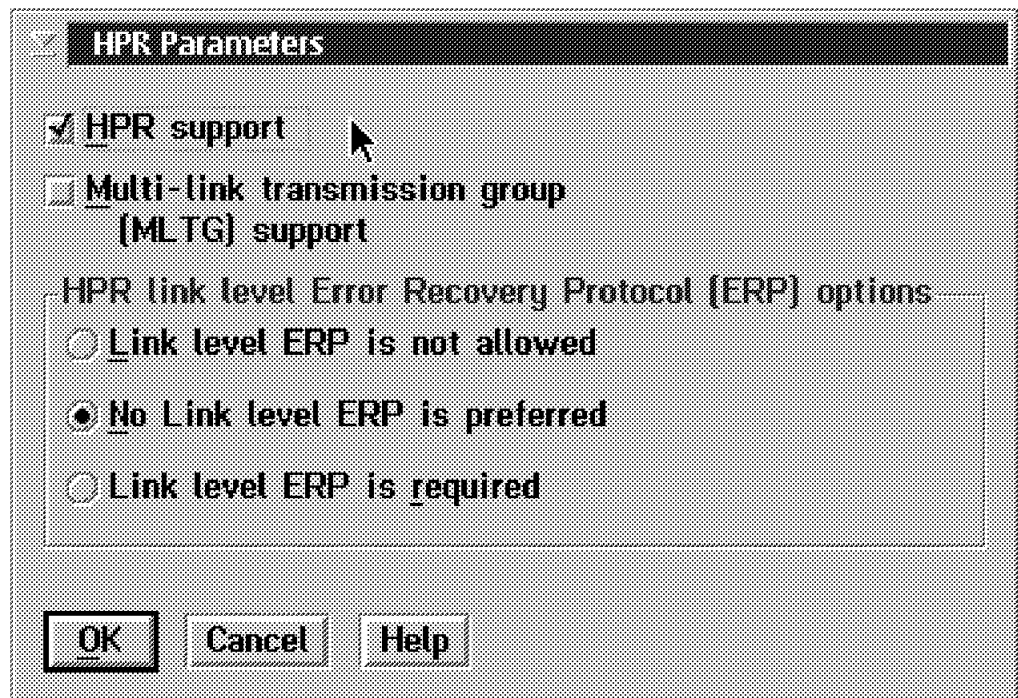


Figure 78. LAN and Frame Relay - DLC HPR Configuration

5.13.1.2 Define Logical Link - HPR Configuration

For explicit links you can override the HPR options that have been previously configured in the DLC profile. Specifically the HPR support (Yes, No, MLTG) and the link level error recovery protocol options as explained before. For link level ERP, the default is to use the options you have previously defined in the DLC profile.

In addition, for multilink TGs, you can select the option to use the default TG number or you can explicitly set it to a value between 1 and 20. All links within the same TG will be assigned the same TG number.

Note

If you change the MLTG TG number, it must match the value you specify in the adjacent node.

☒ **Link Station Protocol Parameters Override**

To override the value specified in the DLC adapter profile for any of the parameters below, select the checkbox, and then enter a value for the parameter.

☐ **Maximum I-field size** [1265-16393]

☒ **Effective capacity (bits per second)** [19200]

☒ **Branch extender support** [Yes]

☒ **HPR support** [MLTG]

HPR link level Error Recovery Protocol (ERP) options

☐ Link level ERP is not allowed

☐ No link level ERP is preferred

☐ Link level ERP is required

☒ Use adapter definition

Multi-link transmission group (MLTG) number

☐ Default TG number

☒ Explicit TG number [10] [1 - 20]

OK Cancel Help

Figure 79. LAN and Frame Relay - Link HPR Configuration

In our scenario we are using TG number 10 for our links. The effective capacity for this link is set to 19.2 kbps as shown in Figure 79. You should configure the

effective capacity to represent a value that is as close as possible to the capacity of your link and consider other factors such as link speed, multidrop links, shared links and so on.

Note

When using MLTGs, HPR will try to use the link with higher capacity first.

5.13.2 SDLC

For implicit links, HPR support and options are configured in the data link control (DLC) profile as shown in Figure 80.

For explicit links, HPR support and options can also be configured in the DLC profile but you can always override the HPR parameters in the connection configuration (define logical link) as shown in Figure 81 on page 135.

In addition, there are some HPR default parameters that you can only change in the node definition file (NDF) configuration as shown in Figure 82 on page 137. For more information on how to install an IBM Wide Area Connector (WAC) adapter for SDLC using HPR and MLTGs, see Chapter 6, "HPR over WAN - SDLC Support" on page 141.

5.13.2.1 DLC HPR Configuration

Select the HPR parameters option in the DLC profile to enable HPR and multilink transmission groups (MLTGs) if required.

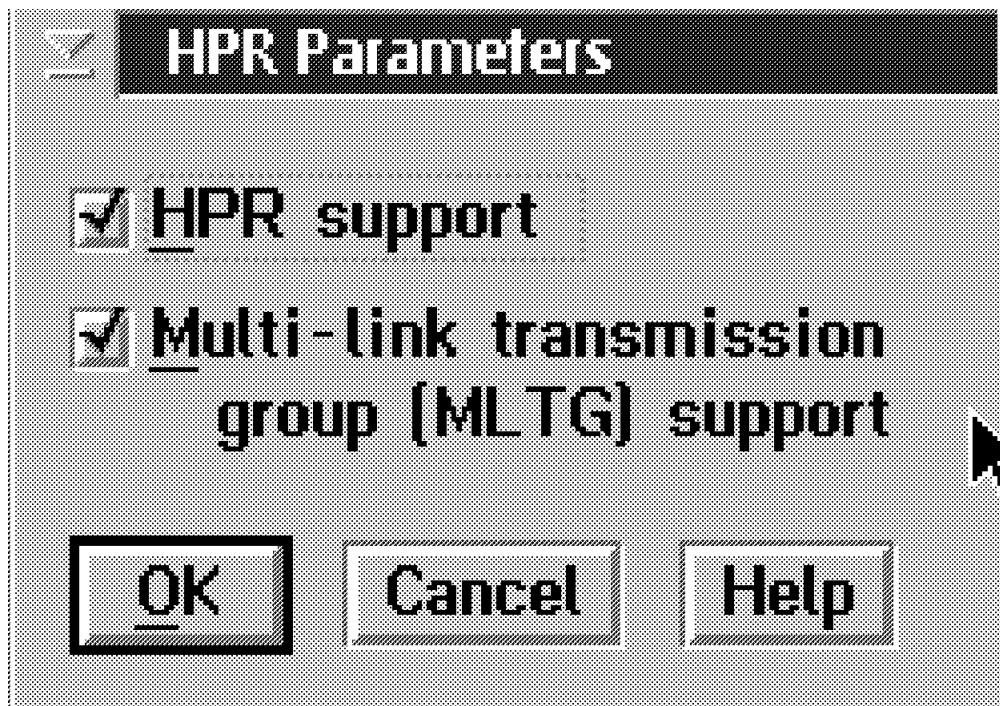


Figure 80. SDLC - DLC HPR Configuration

5.13.2.2 Define Logical Link - HPR Configuration

For explicit links you can override the HPR options that have been previously configured in the DLC profile, specifically the HPR support (Yes, No, MLTG).

Note

Link level error recovery protocol (ERP) options are not implemented in SDLC.

Link Station Protocol Parameters Override for SDLC

To override the value specified in the DLC adapter profile for any of the parameters below, select the checkbox, and then enter a value for the parameter.

☐ Maximum I-field size

☒ Effective capacity [bits per second]

☐ Data transmission mode

☐ Non-XID repoll count

☐ Secondary inactivity timer [seconds]

☐ Primary receive timer [tenths of seconds]

☐ Initialize link with SNRM/SNRME

☐ Branch extender support

☒ HPR support

Multi-link transmission group [MLTG] number

☐ Default TG number

☒ Explicit TG number [1 - 20]

☐ Frame sequence

☒ Module 0 ☐ Module 128

Send window count [1 - 7]

OK Cancel Help

Figure 81. SDLC - Link HPR Configuration

In addition, for multilink TGs, you can select the option to use the default TG number or you can explicitly set it to a value between 1 and 20. All links within the same TG will be assigned the same TG number.

Note

If you change the MLTG TG number, it must match the value you specify in the adjacent node.

In our scenario we are using TG number 10 for our links. The effective capacity for this link is set to 19.2 kbps as shown in Figure 81 on page 135. You should configure the effective capacity to represent a value that is as close as possible to the capacity of your link and consider other factors such as link speed, multidrop links, shared links and so on.

Note

When using MLTGs, HPR will try to use the link with higher capacity first.

5.13.3 NDF Configuration

In the NDF file you will find the logical link definition including the HPR options. The logical link options are also available in the CMSETUP configuration (see Figure 82 on page 137). However, the DLC options are not included in the NDF options and you will only find them in the response file for the configuration.

The NDF file also includes the HPR default values, which are explained in this section. Remember that if you modify the NDF file configuration, you will need to verify the configuration (cmverify command) before it can be used.


```

DEFINE_LOCAL_CP  FQ_CP_NAME(USIBMRA.SERVERB  )
                  CP_ALIAS(serverb  )
                  :
                  BRANCH_EXTENDER_SUPPORT(YES)
                  :
                  ;

DEFINE_LOGICAL_LINK  LINK_NAME(TOSRVAO  )
                    ADJACENT_NODE_TYPE(NN)
                    :
                    DLC_NAME(SDLC  )
                    ADAPTER_NUMBER(0)
                    DESTINATION_ADDRESS(X'01')
                    PERMANENT_CONNECTION_NAME(PCMWACO)
                    CP_CP_SESSION_SUPPORT(YES)
                    :
                    EFFECTIVE_CAPACITY(19200)
                    :
                    HPR_SUPPORT(MLTG)
                    HPR_MLTG_NUMBER(10)
                    BRANCH_EXTENDER_UPLINK(YES)
                    CONNECTION_TYPE(PERMANENT)
                    :
                    MAX_I_FIELD_SIZE(USE_ADAPTER_DEFINITION)
                    :
                    ;

DEFINE_DEFAULTS  IMPLICIT_INBOUND_PLU_SUPPORT(YES)
                 DEFAULT_MODE_NAME(#INTER  )
                 :
                 RETRY_COUNT(6)
                 ALIVE_TIMER(60)
                 PATH_SWITCH_TIMER_LOW(480)
                 PATH_SWITCH_TIMER_MEDIUM(240)
                 PATH_SWITCH_TIMER_HIGH(120)
                 PATH_SWITCH_TIMER_NET(60)
                 ROUTE_SETUP_TIMEOUT(10)
                 MOBILE(NO)
                 :
                 ;

```

Figure 82. HPR - NDF Configuration

- **RETRY_COUNT(6)**

This value indicates how many times HPR will retry an operation after a link failure. When this value is reached, HPR looks for an alternate path. Default value is 6 times.

- **ALIVE_TIMER(60)**

If there is no HPR traffic for the time specified in seconds, HPR sends an alive packet to the remote node to indicate that the connection is still available. Default value is 60 seconds.

- **PATH_SWITCH_TIMER_LOW(480)**

If a particular link fails, the sending node looks for a new path for the maximum time specified. This value is for low priority class of service (COS #BATCH) connections. Default value is 480 seconds (8 minutes).

- **PATH_SWITCH_TIMER_MEDIUM(240)**
If a particular link fails, the sending node looks for a new path for the maximum time specified. This value is for medium priority class of service (COS #CONNECT) connections. Default value is 240 seconds (4 minutes).
- **PATH_SWITCH_TIMER_HIGH(120)**
If a particular link fails, the sending node looks for a new path for the maximum time specified. This value is for high priority class of service (COS #INTER) connections. Default value is 120 seconds (2 minutes).
- **PATH_SWITCH_TIMER_NET(60)**
If a particular link fails, the sending node looks for a new path for the maximum time specified. This value is for network priority class of service (COS #SNASVCMG) connections. Default value is 60 seconds (1 minute).
- **ROUTE_SETUP_TIMEOUT(10)**
If the ROUTE_SETUP Control Vector (RSCV) is not received at the requesting node within 10 seconds, the route setup command times out and is acknowledged as having no route to the destination. The connection is not established.
- **MOBILE(NO)**
This option indicates if the user of this connection is mobile or not. That is, it indicates if the user has temporarily disconnected from the network with his or her computer.

If you configure MOBILE(YES) and the user has disconnected from the network without explicitly terminating connections, then the HPR connections are not terminated. When the user returns and reconnects the computer to the network, the old connections will still be active. Hence there will be no need, for example, to LOGIN again, into a system that the user previously had logged in to.

5.14 Monitor HPR Connections

eNetwork Communications Server for OS/2 Warp provides two commands (CMCONNS and HPRCONNS) you can use to monitor the HPR connections. You can also invoke the subsystem management utility to display information about HPR activity.

5.14.1 HPR Commands

From the command line you can issue the following commands:

- **CMCONNS.** It displays RTP connections.
- **HPRCONNS.** It invokes CMCONNS and give you additional options.

5.14.1.1 CMCONNS

CMCONNS can be used to view and monitor your HPR connections including intermediate routing nodes in the network. It can also be used to force HPR to calculate the best path for a specific connection. The command line syntax is shown below for the CMCONNS command.

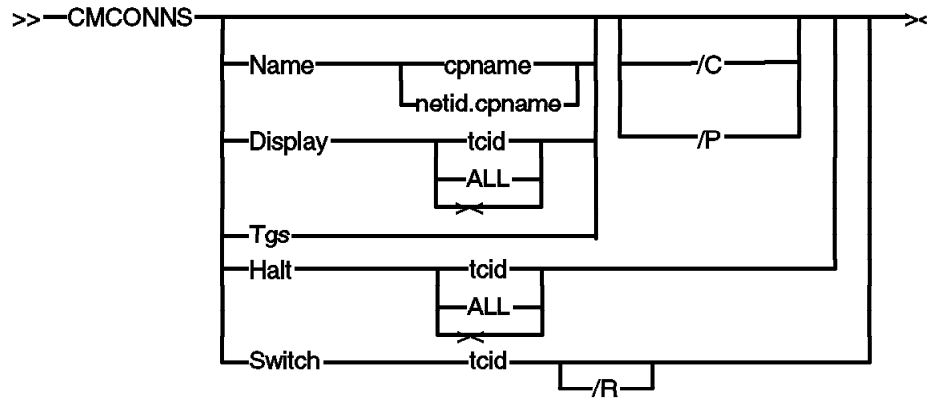


Figure 83. CMCONNS Command Line Syntax

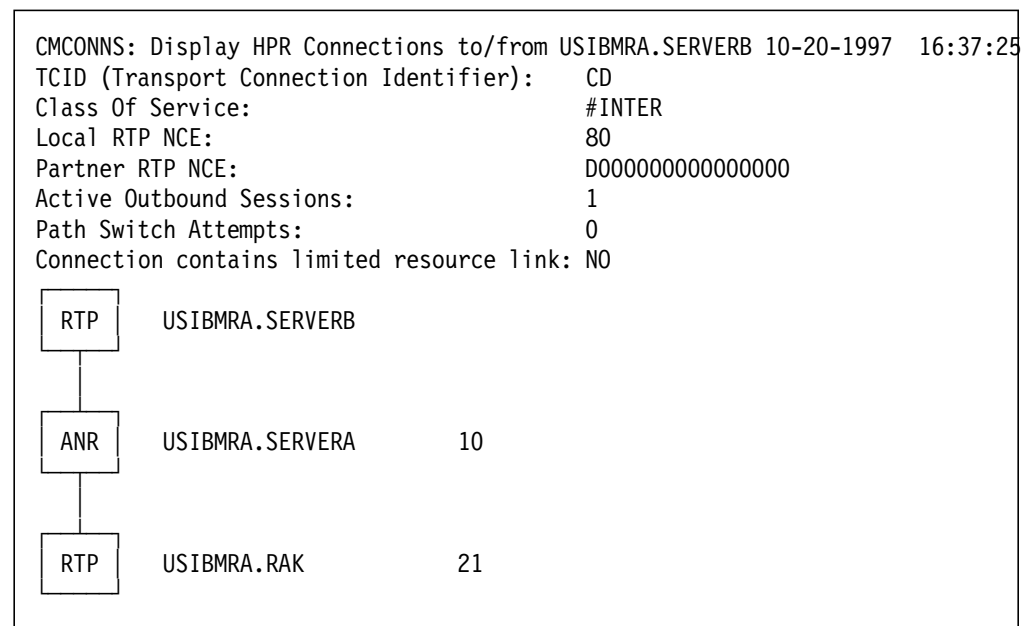


Figure 84. CMCONNS - Sample Display

5.14.1.2 HPRCONNS

HPRCONNS is a REXX program that invokes CMCONNS; see Figure 83. It can be used to view and monitor your HPR connections including intermediate routing nodes in the network. It can also be used to force HPR to calculate the best path for a specific connection. HPRCONNS waits for you to go to the next page. The command line syntax is shown below for the HPRCONNS command.

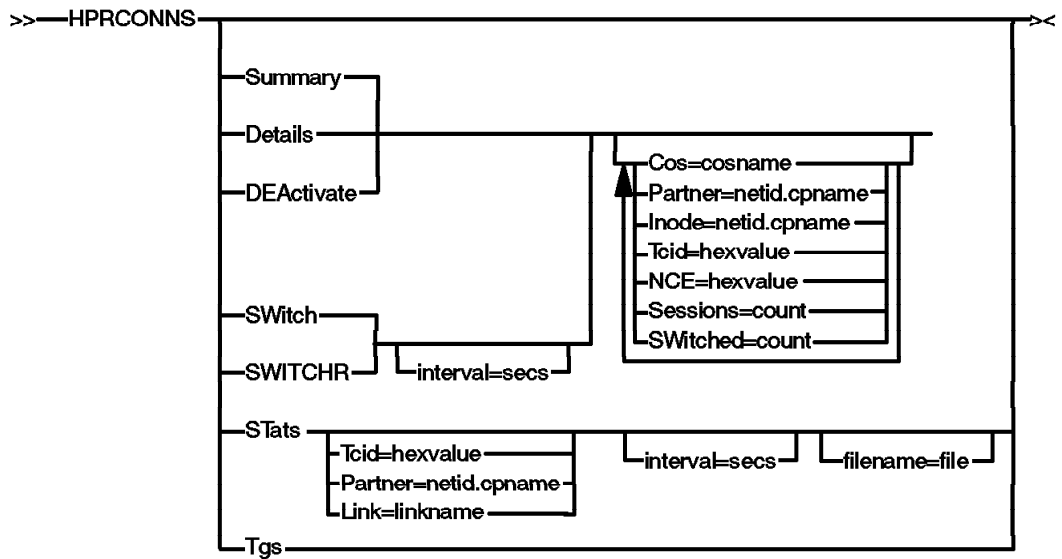


Figure 85. HPRCONNS Command Line Syntax

The following is the output from the command HPRCONNS:

```

CMCONNS: Display HPR Connections to/from USIBMRA.SERVERB 10-20-1997 16:38:06
TCID (Transport Connection Identifier):  CD
Class Of Service:                        #INTER
Local RTP NCE:                           80
Partner RTP NCE:                         D000000000000000
Active Outbound Sessions:                 1
Path Switch Attempts:                     0
Connection contains limited resource link: NO

  RTP    USIBMRA.SERVERB
  |
  ANR    USIBMRA.SERVERA    10
  |
  RTP    USIBMRA.RAK        21

Press ENTER to display the next connection.
  
```

Figure 86. HPRCONNS - Sample Display

In Figure 86 we can read information about active RTP connections including TG numbers used (10 and 21) by the links, number of sessions for the specific class of service (COS), CP names of the nodes and the HPR function (RTP or ANR).

Chapter 6. HPR over WAN - SDLC Support

eNetwork Communications Server for OS/2 Warp Version 5.0 has implemented SDLC connections to support HPR over WAN. This is a new feature in this version and was not present in previous versions of the product. In this chapter we describe the steps required to configure the SDLC support in eNetwork Communications Server for OS/2 Warp for HPR over WAN and using the IBM Wide Area Connector (WAC) adapter.

6.1 Installation of SDLC Protocol

In this section we explain how we set up the SDLC connections for Figure 87.

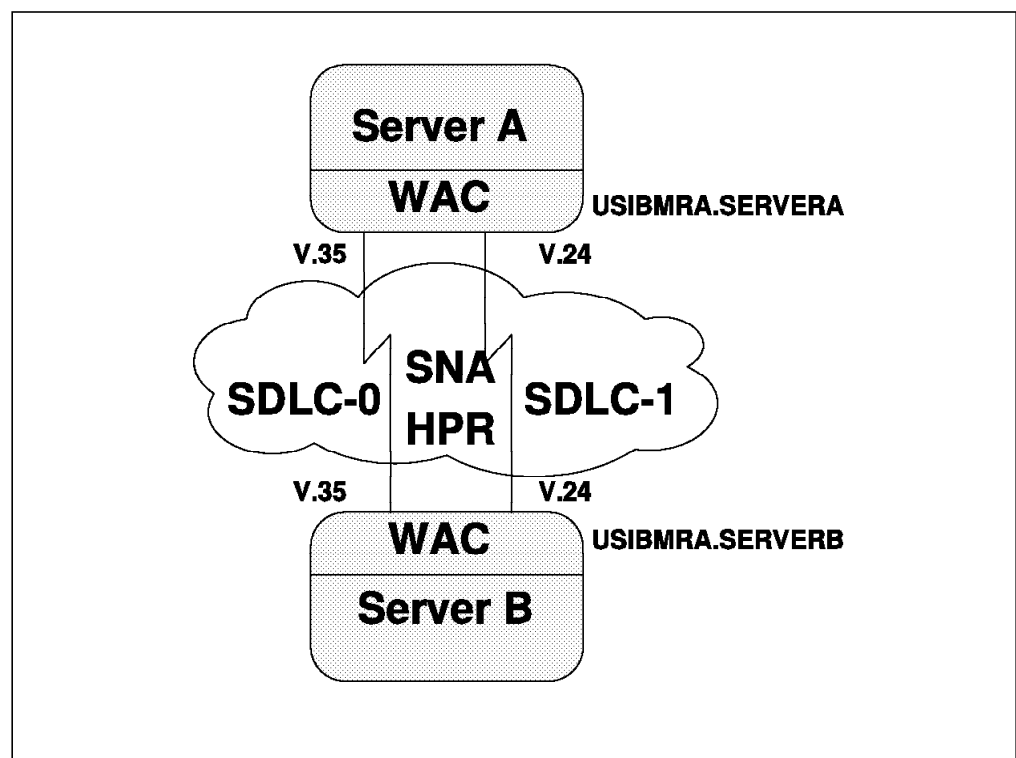


Figure 87. SDLC Scenario for HPR over WAN

The two servers, Server A and Server B are connected together via two 19.2kbps SDLC lines: SDLC-0 and SDLC-1. SDLC-0 is via a V.35 external clocking connection and the SDLC-1 is via a V.24 internal clocking connection.

Note: Server B is a branch extender node in this configuration with HPR over WAN support.

The recommended sequence of steps for SDLC protocol support installation and configuration is as follows:

- In MPTS:
 1. Install WAC adapter card driver support in MPTS. This step will include the physical drivers to be available in MPTS configuration.
 2. Select WAC adapter port drivers (for SDLC lines) in MPTS. This step will make the WAC drivers available in your system (config.sys).

3. Configure IBM WAC parameters for each port. The WAC adapter options must be configured to use the Advanced Network Device Interface Specification (ANDIS) which is required when using the SDLC protocol provided by eNetwork Communications Server for OS/2 Warp. The configuration values go to the protocol.ini file.
 4. Save and exit MPTS configuration.
 5. Reboot your system.
- In eNetwork Communications Server for OS/2 Warp (CMSETUP):
 1. Configure SNA Local Node configuration. This step is mandatory to provide a network ID and CP name to your server.
 2. Configure SNA Port Connection Manager. A PCM is configured for hooking into MPTS.
 3. Configure DLC SDLC configuration.
 4. Configure SNA connections (define logical links).

6.1.1 WAC Adapter - Installation

In this section we explain the required steps to install the WAC adapter drivers. In our scenario, the ISA WAC drivers are installed from the eNetwork Communications Server for OS/2 Warp CD-ROM.

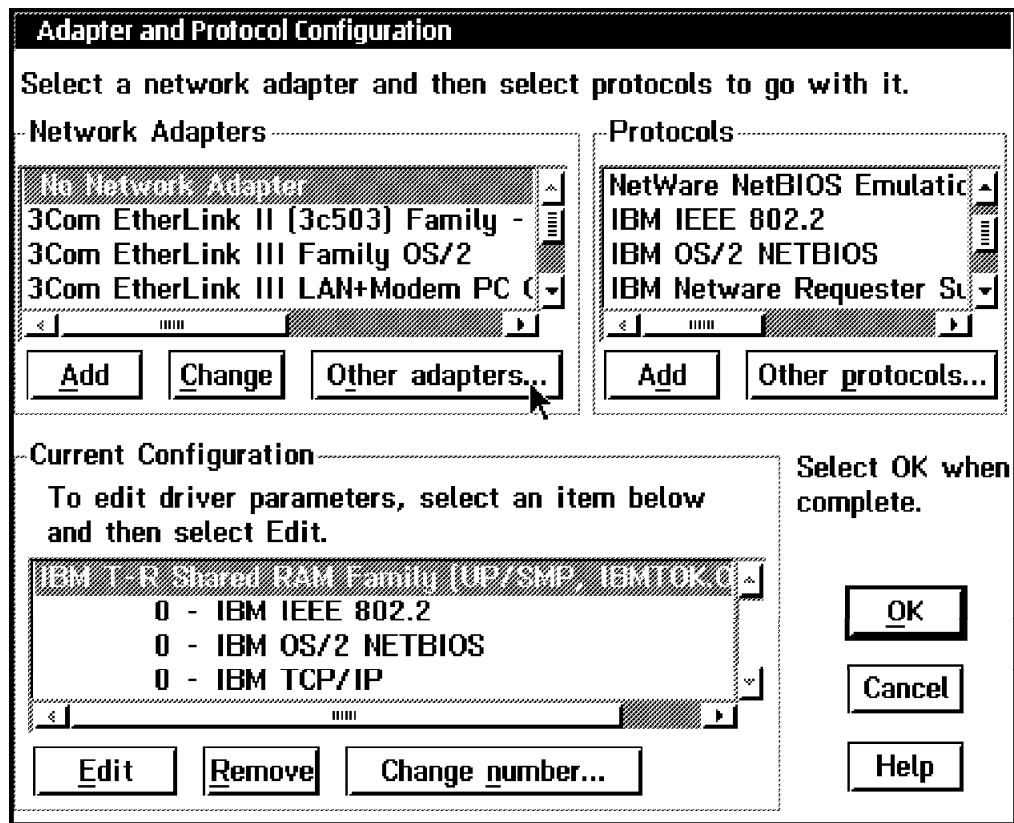


Figure 88. WAC Adapter Installation - MPTS

The following steps are required to install the WAC adapter drivers:

- Add WAC adapter driver to MPTS.

- WAC can be either MCA, ISA or PCI.
- Select Other adapters...
- Insert diskette or CD (\drivers\ibmwac\).

6.1.1.1 ISA Bus Hardware Installation

The ISA WAC card may be put into any free 16-bit slot. It is important, however, that the I/O address range settings (switches 4-7) of the WAC adapter not duplicate those addresses of any other device in your system. If further assistance is required to set the switch positions, your local systems coordinator should be consulted.

When configuring the ISA WAC card, find the Logical Card Number in the chart below which corresponds to your card's switch or address settings. Make a note of this logical card number.

I/O Address (Hex)	Logical Card Number	Dip Switch			
		4	5	6	7
0120 - 013F	15	ON	ON	ON	ON
0140 - 015F	14	OFF	ON	ON	ON
0180 - 019F	13	ON	OFF	ON	ON
01A0 - 01BF	12	OFF	OFF	ON	ON
0220 - 023F	11	ON	ON	OFF	ON
0240 - 025F	10	OFF	ON	OFF	ON
0280 - 029F	9	ON	OFF	OFF	ON
0280 - 029F	8	OFF	OFF	OFF	ON
0520 - 053F	7	ON	ON	ON	OFF
0540 - 055F	6	OFF	ON	ON	OFF
0580 - 059F	5	ON	OFF	ON	OFF
05A0 - 05BF	4	OFF	OFF	ON	OFF
0620 - 063F	3	ON	ON	OFF	OFF
0640 - 065F	2	OFF	ON	OFF	OFF
0680 - 069F	1	ON	OFF	OFF	OFF
0680 - 069F	0	OFF	OFF	OFF	OFF (all up)

Note: In OS/2 you need to use the MPTS configuration utility to configure the adapter. Enter the Logical Card Number in the MPTS configuration field titled Card Number.

In a NetWare server the logical card number is specified by the SLOT parameter on the LOAD line for WACSDLC.

Note

When the WAC adapter is used in IBM PS300 Models 657x and 658x (where x can be any number), the adapter must be set to use interrupt 9 only. The use of any other interrupt may cause erratic behavior due to lost interrupts.

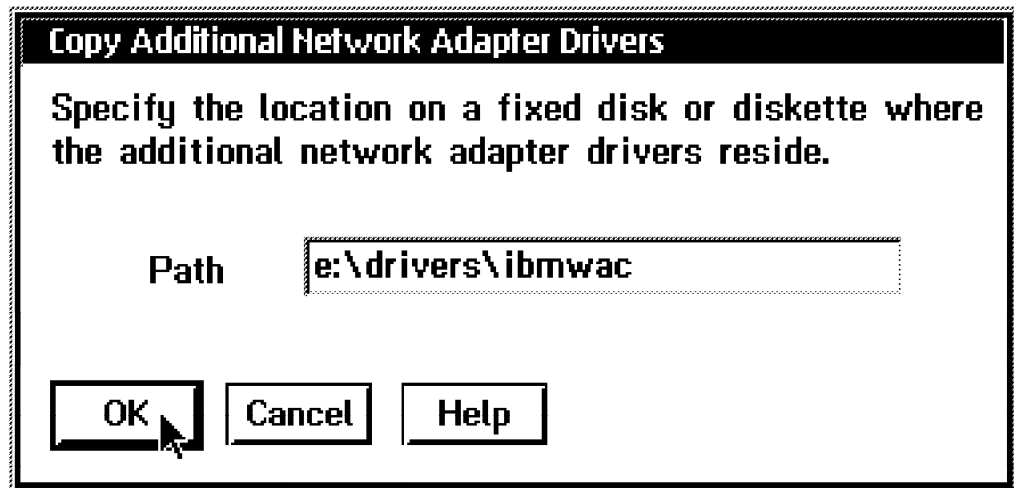


Figure 89. WAC Driver Installation

6.1.1.2 Device Driver Installation

The OS/2 IBM WAC device driver conforms to the IBM extensions of the 3Com/Microsoft Local Area Network (LAN) Manager Network Driver Interface Specifications (NDIS) Version 2.01 (Final).

The following procedure will install the driver:

1. Open an OS/2 window for full screen.
2. Run the MPTS installation and configuration program by changing to the directory where the MPTS.EXE file resides (typically in C:\IBMCOM).
3. Enter MPTS at the command prompt.
4. Insert the WAC installation and testing diskette into your diskette drive. Select **Install** from the MPTS main menu.
5. Select **Additional Network Drivers** from the Installation menu, select **Continue...** and press Enter.
6. Modify the screen to match your diskette drive letter and the path to the OS/2 MAC driver. Typically, this is A:\OS2. In the Communications Server CD this is d:\drivers\ibmwac.
7. Select OK and press Enter. MPTS will now copy the OS/2 device driver onto your hard drive

a:\os2\IBMWAC.OS2	into	C:\IBMCOM\MACS directory
a:\os2\IBMWAC.NIF	into	C:\IBMCOM\MACS directory
a:\os2\WAC.MSG	into	C:\IBMCOM directory
a:\os2\WACH.MSG	into	C:\IBMCOM directory
8. After the device driver files have been successfully transferred, exit to the MPTS main menu.

The WAC driver is now installed. In most cases, your next step will be to install or configure your specific communications options.

6.1.1.3 Install Two WAC Ports for Two SDLC lines

From the Network Adapters window add the IBM Wide Area Connector to the Current Configuration window. Add a second WAC port to the Current Configuration window if you are planning to use both ports of the WAC adapter. In our scenario we have two SDLC lines between Server A and Server B.

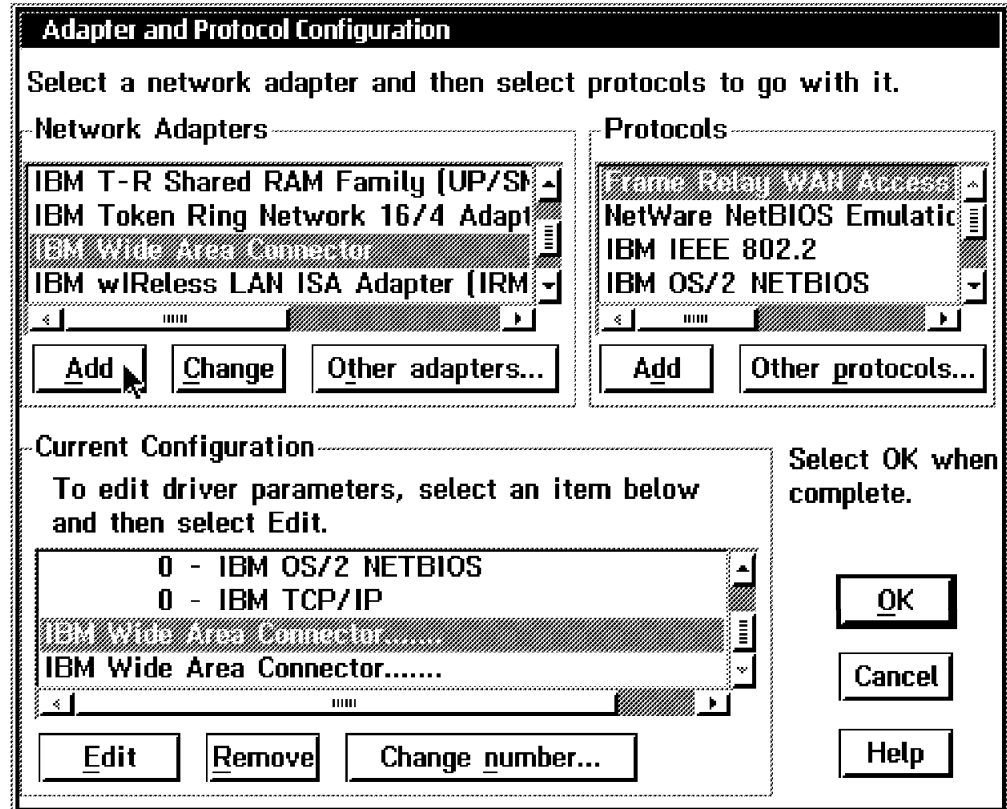


Figure 90. Installation of WAC Adapter Driver - Upper and Lower Ports

6.1.1.4 Configure IBM WAC Parameters

The following configuration guidelines can be used to configure a WAC adapter port in order to support SDLC.

1. Enter the slot number. For example, enter 15 as the slot number if the DIP switches on your ISA WAC adapter card are all in the off position.

Parameters for IBM Wide Area Connector
Edit the parameters as needed.

Slot number [MC-A] [1..8] or Card Number [ISA] [0..15]	15
Port number [0 - upper, 1 - lower]	0
16KB Shared memory addr. [0C0000-FE0000]	0D0000
Line Mode [0 - Constant RTS, 1 - Switched RTS]	0
NRZI [0 - No, 1 - Yes]	0
RS232/V.24 mode [0 -DTE, 1 -DTE_Pin24_TxC]	0
RS422 mode [0] or X.21 Mode [1-4]	0
Line Speed in bits/second	19200
Maximum frame size to be sent and received	4486
MAC Type Description	HDLC
Maximum number of outstanding transmit requests	8
ANDIS PCM support [0 - No, 1 - Yes]	1
Link connection type [0 - leased, 1 - switched]	0

OK Range Cancel Help

Figure 91. Configuration Parameters of WAC Card - Port Zero (Upper)

2. Enter Port number as 0 for the upper port on the WAC card. Enter Port number as 1 for the lower port on the WAC card.
3. Enter 1 for ANDIS PCM support.

Note

SDLC protocol in eNetwork Communications Server for OS/2 Warp requires ANDIS PCM support.

4. Enter 0 for RS232/V.24 mode if your modem does not require internal clocking.

Parameters for IBM Wide Area Connector
Edit the parameters as needed.

16KB Shared memory addr. [0C0000-FE0000]	0D0000
Line Mode [0 - Constant RTS, 1 - Switched RTS]	0
NRZI [0 - No, 1 - Yes]	0
RS232/V.24 mode [0 -DTE, 1 -DTE_Pln24_TxC]	0
RS422 mode [0] or X.21 Mode [1-4]	0
Line Speed in bits/second	19200
Maximum frame size to be sent and received	4486
MAC Type Description	HDLC
Maximum number of outstanding transmit requests	8
ANDIS PCM support [0 - No, 1 - Yes]	1
Link connection type [0 - leased, 1 - switched]	0
Portname displayed in the PPAT table	WAC0
Minimum number of flags between transmit frames	1

OK Range Cancel Help

Figure 92. WAC Port Configuration Parameters

5. Enter the port name. For example, WAC0. This name will be used later in the port connection manager (PCM) configuration.

6.1.2 Configure SDLC Protocol

After installing the WAC adapter and driver and after you have configured the WAC ports in MPTS for SDLC support, you will now configure eNetwork Communications Server for OS/2 Warp to support HPR and multilink TGs (MLTGs) over SDLC.

6.1.2.1 Port Connection Manager Configuration

For SDLC, you are required to configure a PCM. From the communications server configuration list window, select SNA Phone Connect - Port Connection Manager to configure a PCM as shown in Figure 93 on page 148.

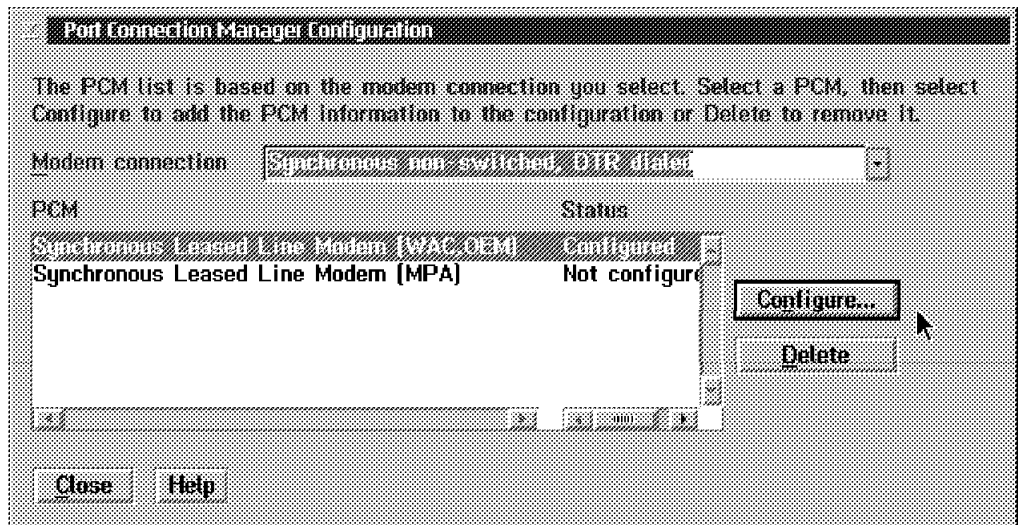


Figure 93. Port Connection Manager Configuration for SDLC over WAC

Note: A PCM is a program that serves as an interface between the connection manager and your communications adapter to manage your connections. Each adapter on your workstation has a PCM that is provided by the adapter manufacturer. When you configure a PCM, you set it up for connection manager use.

The PCM is used to define parameters to Communications Server for Synchronous Leased Line Modems that are installed in your machine. This profile is required for nodes that establish SNA phone connections.

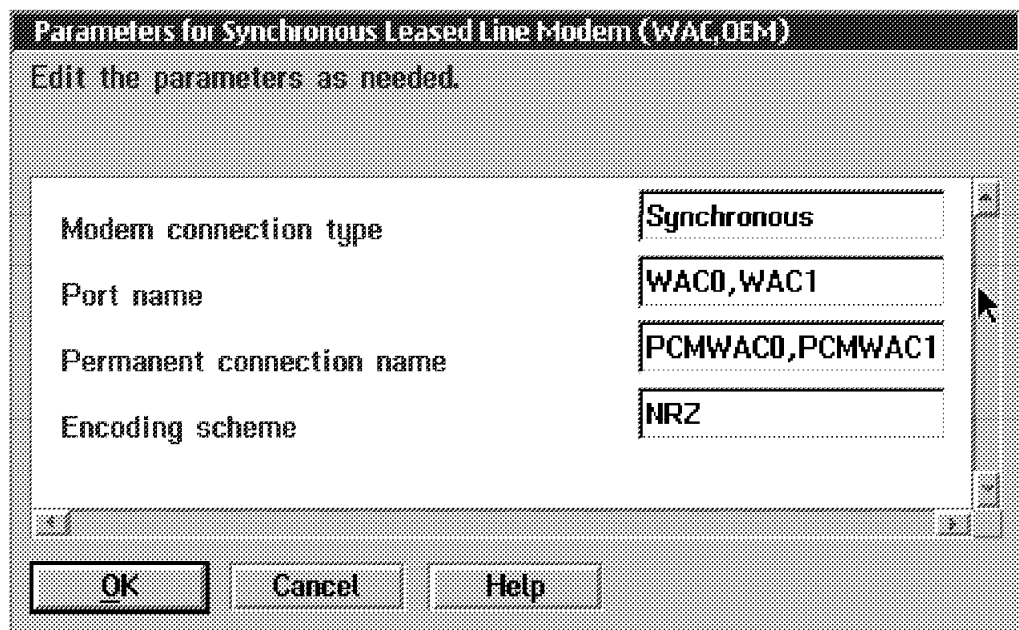


Figure 94. PCM Configuration for 2 WAC Ports (WAC0 and WAC1)

You will configure the PCM parameters as required by your installation. For example:

1. Modem connection type is synchronous.

2. Enter the Port name. This parameter identifies the Media Access Control (MAC) for the communication hardware port this Port Connection Manager (PCM) will use. If you want this PCM to manage multiple ports, specify the name of each port here, separated by a comma, for example, WAC0, WAC1.

Note

The name or names you specify here must match the name or names you have configured for the MAC in MPTS.

3. Enter the connection name. This parameter enables you to specify the permanent connection name associated with the port or ports that this Port Connection Manager (PCM) will manage. For each port specified in the Port Name field, enter the permanent connection name for that port in this field, separated by comma, for example, PCMWAC0, PCMWAC1. The Permanent Connection Name specified here can be the same as the Port name but a different name will help in problem determination.
4. Select the encoding scheme (for example, NRZ versus NRZI).

6.1.2.2 SDLC Adapter Profile

For each WAC port using SDLC, you are required to configure an SDLC adapter profile. Some of the options you configure here can be overridden in the logical link definition (for example the HPR parameters).

From the Communications Server Configuration List window, select SDLC DLC to configure the SDLC adapter profile as shown in Figure 95.

SDLC DLC Adapter Parameters

Adapter: 0 [0 - 127]

☒ User-controlled dialing

☒ Branch extender support

Maximum l-field size: 1033 [265 - 4105]

SNA Phone Connection parameters

☒ Accept incoming calls

Incoming call directory entry: []

Secondary station address (hex): 1 [01 - FE]

Link station role: Negotiable

Effective capacity (bits per second): 19200

Additional parameters:

- HPR parameters
- Link deactivation parameters
- Link initialization parameters
- Link station protocol parameters
- Resource parameters

Change...

Optional comment: []

OK Delete Cancel Help

Figure 95. SDLC Adapter Configuration - Adapter 0

Enter the Effective Capacity for this profile. In our scenario, it is estimated to be 19.2 kbps. Note that this figure only represents a weighting factor and does not control the speed of the SDLC line in any way. The actual speed of the line may be different in many cases, for example, when the line is shared with other protocols or it is a multidrop line.

You will now select HPR parameters and click Change... to include HPR and MLTG support as indicated in Figure 96.

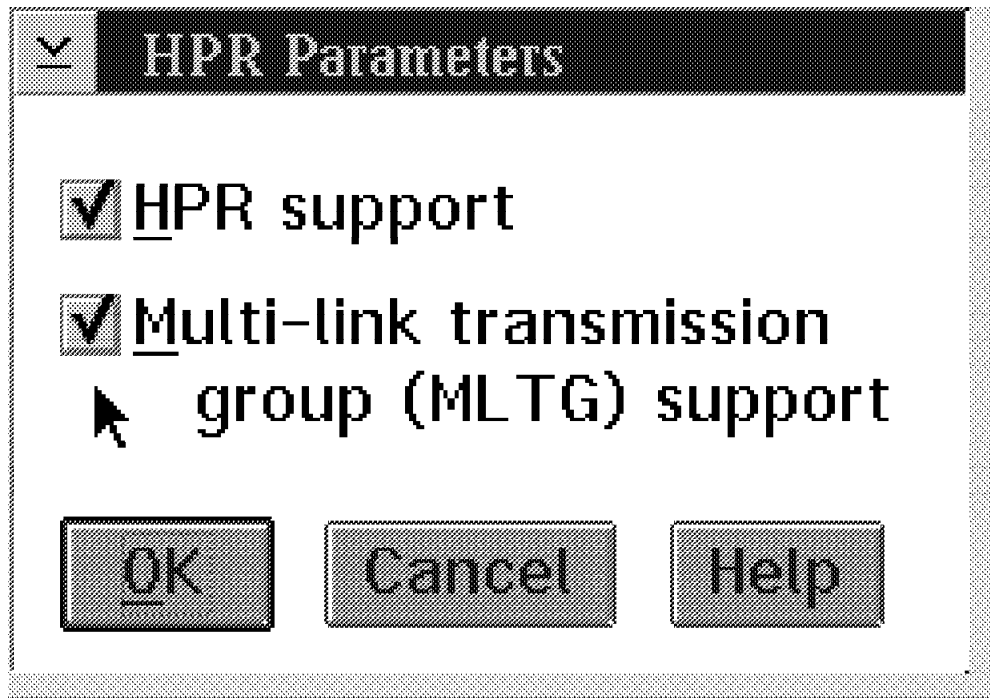


Figure 96. HPR and MLTG Support for SDLC

By default we want to support HPR and MLTGs on SDLC links assuming that the adjacent APPN node supports HPR and it may use more than one link for connection on the same transmission group (TG). If HPR is not supported by the adjacent node then it defaults back to intermediate session routing (ISR). In other words, the HPR options are negotiated at XID3 exchange when the link between the two nodes is established.

6.1.2.3 Define Logical Links

You will explicitly define connections to adjacent nodes using the previously defined SDLC profiles. When defining the logical link parameters you have the option to override the HPR parameters you configured in the SDLC profile if any. In our scenario, two links or connections are defined to the same adjacent node using PCMWAC0 and PCMWAC1 and the same TG number 10.

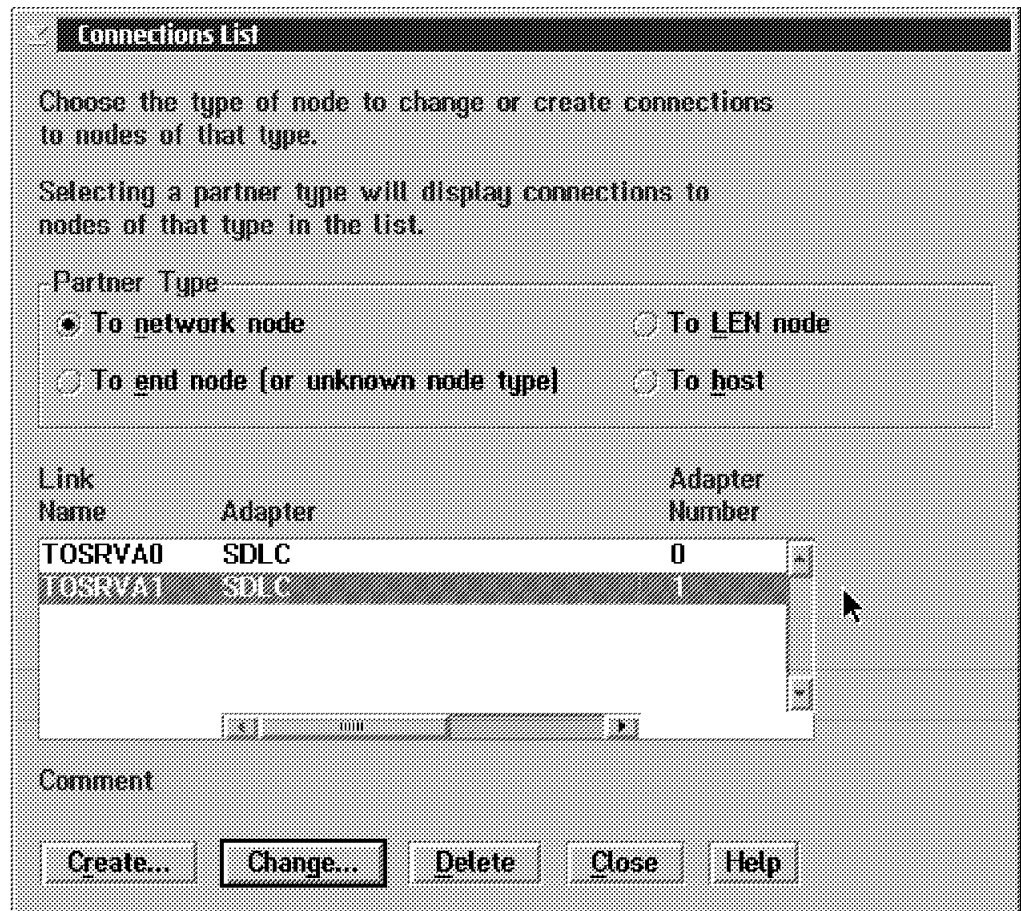


Figure 97. WAC SDLC Connections

You will select SDLC and Adapter number. In our scenario, there are two connections on two different adapter numbers (0 and 1) but both links will be using the same transmission group (TG) number 10.

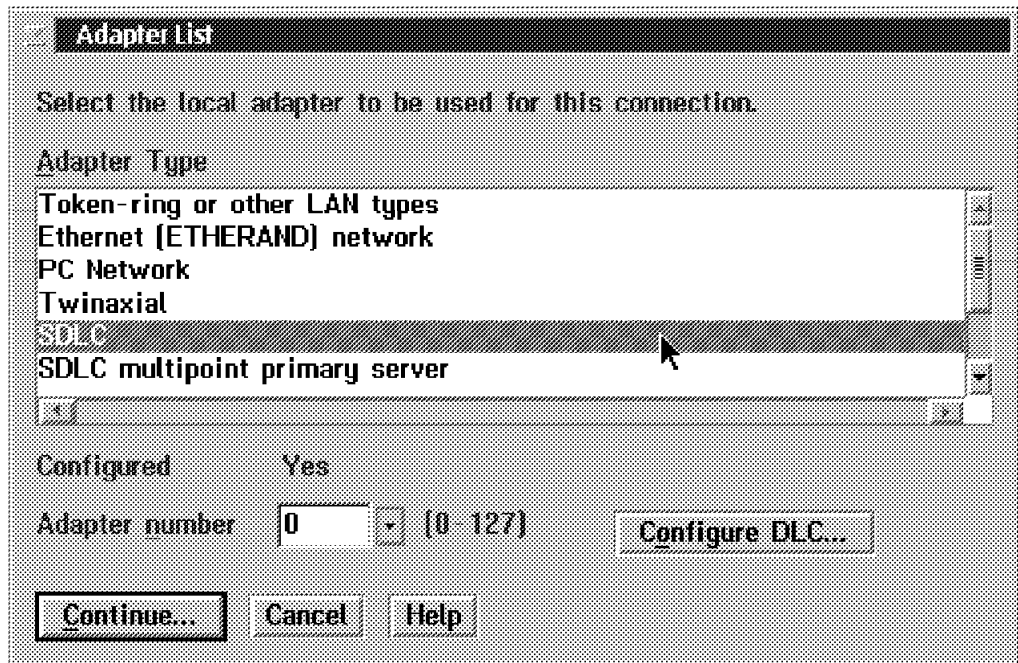


Figure 98. Define Logical Link - Selecting SDLC DLC

Next, you will enter the SDLC connection options as indicated in Figure 99 on page 153. For example:

- For identification purposes, enter the name of the connection. For example TOSRVA0 indicates it is a connection to server A using SDLC port 0 and TOSRVA1 indicates it is a connection to server A using SDLC port 1.
- The SDLC secondary station address we are using is 0x01.
- The connection type is permanent since we are simulating leased lines in this configuration.
- The permanent connection name (PCM) is selected on port zero as it was previously defined (PCMWAC0 for port 0 and PCMWAC1 for port 1).
- Use the Override... option to redefine parameters entered in the SDLC profile for this connection (see Figure 99 on page 153).

Connection to a Network Node

Link name: TOSRVA0 ☒ Activate at startup

Adjacent node ID (hex):

Partner LU definitions

Partner network ID: Define Partner LUs...

Partner node name:

Secondary station address (hex): 01 [01-FE]

SNA Phone Connect parameters

Connection type: Permanent

Permanent connection name: PCMWAC0

Outgoing call directory entry:

To provide unique link protocol parameters that are different than those specified in the DLC adapter profile, select Override...

Override...

OK Additional parameters... Cancel Help

Figure 99. SDLC Connection to a Network Node (NN)

By using the override option, you can redefine some parameters previously defined in the SDLC profile. For HPR over SDLC, we configure the following parameters as seen in Figure 100 on page 154:

- Effective capacity: 19.2 kbits per second
- HPR support: MLTG
- MLTG explicit TG number: 10

Note

If you enter a TG number, it must match the TG number in the adjacent node.

- In our scenario, server B is a BrNN so we enable this uplink to have branch extender support.

Link Station Protocol Parameters Override for SDLC

To override the value specified in the DLC adapter profile for any of the parameters below, select the checkbox, and then enter a value for the parameter.

☐ Maximum I-field size		(265 - 4105)
☒ Effective capacity (bits per second)	19200	
☐ Data transmission mode		
☐ Non-XID repoll count		(1 - 127)
☐ Secondary inactivity timer (seconds)		(40 - 160)
☐ Primary receive timer (tenths of seconds)		(10 - 200)
☐ Initialize link with SNRM/SNRME		
☒ Branch extender support	Yes	
☒ HPR support	MLTG	

Multi-link transmission group (MLTG) number

☐ Default TG number

☒ Explicit TG number (1 - 20)

☐ Frame sequence

☒ Modulo 8 ☐ Modulo 128

Send window count (1 - 7)

Figure 100. Define Logical Link - SDLC Override Options

6.2 Troubleshooting

Make sure the WAC adapter drivers have been successfully loaded by displaying lantran.log. It will normally look like this:

```

Card: 15 Port: 0 Hardware Interface: V35
I/O Port: 0120 Irq: 9 Shared Ram: 000D0000
Rx Clock: External Tx Clock: External
Card: 15 Port: 1 Hardware Interface: RS232
I/O Port: 0128 Irq: 9 Shared Ram: 000D0000
Rx Clock: Internal Tx Clock: Internal
IBMWAC 2.01.00 successfully loaded.

```

Figure 101. WAC Adapter Initialization - LANTRAN.LOG

You will also monitor your SDLC connections using either subsystem management or the cmlinks command to display your connections. For APPN connections, you will also have to make sure you have CP-CP sessions. CP-CP sessions can be easily identified by the mode name CPSVCMG.

In our scenario we can see that we are using MLTGs with TG number 10 for SDLC ports 0 and 1 as indicated in the following display:

Link Name	DLC Name	#	Partner FQName	TG#	Type	HPR	Sess
TOSRVA0	SDLC	0	USIBMRA.SERVERA	10	EN	MLTG	MLTG
TOSRVA1	SDLC	1	USIBMRA.SERVERA	10	EN	MLTG	2

Figure 102. SDLC Connections Using MLTG Number 10

Chapter 7. Branch Extender Node

Advanced Peer-to-Peer Networking is a set of functions, formats, and protocols that greatly enhance the managing of an SNA network and the usability of APPC applications running in the network. APPN does this through reduced configuration requirements, dynamic directory searches, route calculation capabilities, and intermediate session routing.

APPN has been enhanced to support branch extender nodes (BrNN). In this chapter we describe the new architecture as implemented in eNetwork Communications Server for OS/2 Warp. A sample configuration is also included.

7.1 Overview

Advanced Peer-to-Peer Networking (APPN) is an extension of SNA that adds communications functions. Its basic components include:

- APPN node types
- Control points
- Branch extender
- Data link control
- Logical links
- Directory services
- Topology and route-selection services

With APPN, you can write programs without knowing the details of the underlying network. All you need to know is the name of the partner LU; you do not need to know its location. SNA determines the partner LU location and the best path for routing data. A change to the underlying network, such as a physical address change, the addition of a new adapter, or the relocation of a machine, does not affect APPC programs.

The branch extender is a border node subset that is designed to interconnect a branch office to an APPN WAN backbone network. The Branch Extender function provides optimized peer-to-peer communication, so large APPN network customers can connect many LAN-based branches via one large WAN, primarily based on switched links. It addresses having too many network nodes in the topology database. It adds the ability to register resources from a branch to a central directory server in the WAN. In conjunction with APPN switched connection networks, it also makes it possible to have direct single-hop connections between two APPN systems in different APPN topology subnetworks which are on the same switched connection network.

A BrNN is a NN which typically has LAN and WAN interfaces. To downlink devices (downlinks, typically LAN), a BrNN looks like an APPN NN. To uplink devices, a BrNN looks like an APPN EN. The BrNN is not a new APPN node type. It appears as a NN or EN, depending upon where the viewer is in the network. BrNN is a combination of EN and NN. There are the advantages of Branch Extender:

- Allows very large APPN networks
- Complimentary to border node:

- border node for large intranets requiring security, accounting
- branch extender for large intranets with many branches
- Interconnects branch sites to APPN WAN backbone
- Acts like EN to APPN backbone
- Acts like NN server to branch workstations
- Hides branch topology from WAN, reducing overhead on WAN traffic

The Branch Extender enables you to interconnect a branch office with LANs, end nodes and low end network nodes with dependent and independent LUs, and PUs such as teller machines, to one or several WANs. The Branch Extender enhances performance in large APPN networks. Specifically, it:

- Reduces the number of network nodes in large APPN networks, enabling you to add additional branch networks
- Hides branch topology information
- Enables peer-to-peer communication between branches connected to the same APPN connection network
- Enables coexistence with PU gateway servers
- Reduces uplink CP-CP session traffic (WAN traffic)
- Isolates the branch network from backbone WAN traffic overhead

Branch Extender support reduces CP-to-CP traffic, because TDU and LOCATE flows are no longer required.

The Peripheral Border Node architecture was designed to connect two different APPN networks with different netIDs by acting as a broker. It will show its EN face to the NN with the different netID and its NN face to the node with the same network ID. The major similarity between PBN and Branch Extender is that both show an EN image on certain links while showing an NN image on other links. The major difference is that while PBN hides both networks from each other, Branch Extender makes no attempt to conceal the uplink network from the branch (domain) network. From domain nodes' point of view, they are connected to the uplink network via a regular NNS, which connects to other NNSs using normal CP-CP sessions.

Note

eNetwork Communications Server for OS/2 Warp Version 5.0 cannot be configured as a peripheral border node.

Branch extender and border nodes are complementary.

- Branch extender is the best solution for large intranets with many branches.
- Border nodes are the best solution for internets.
- Branch extender and border nodes can be mixed for an optimal network design.

Note

eNetwork Communications Server for OS/2 Warp Version 5.0 cannot be configured as a border node.

7.1.1 Necessity

Without the Branch Extender, problems occur if an APPN topology subnetwork exceeds the smallest network node's topology database (TDB) storage. This could limit network size to as few as 150 network nodes. TDB size is a function of the number of network nodes and links, as well as the frequency of state changes. Many administrators would prefer to install 1500 or more small network nodes as gateways to branch offices, with a LAN at each branch office that is configured as a connection network.

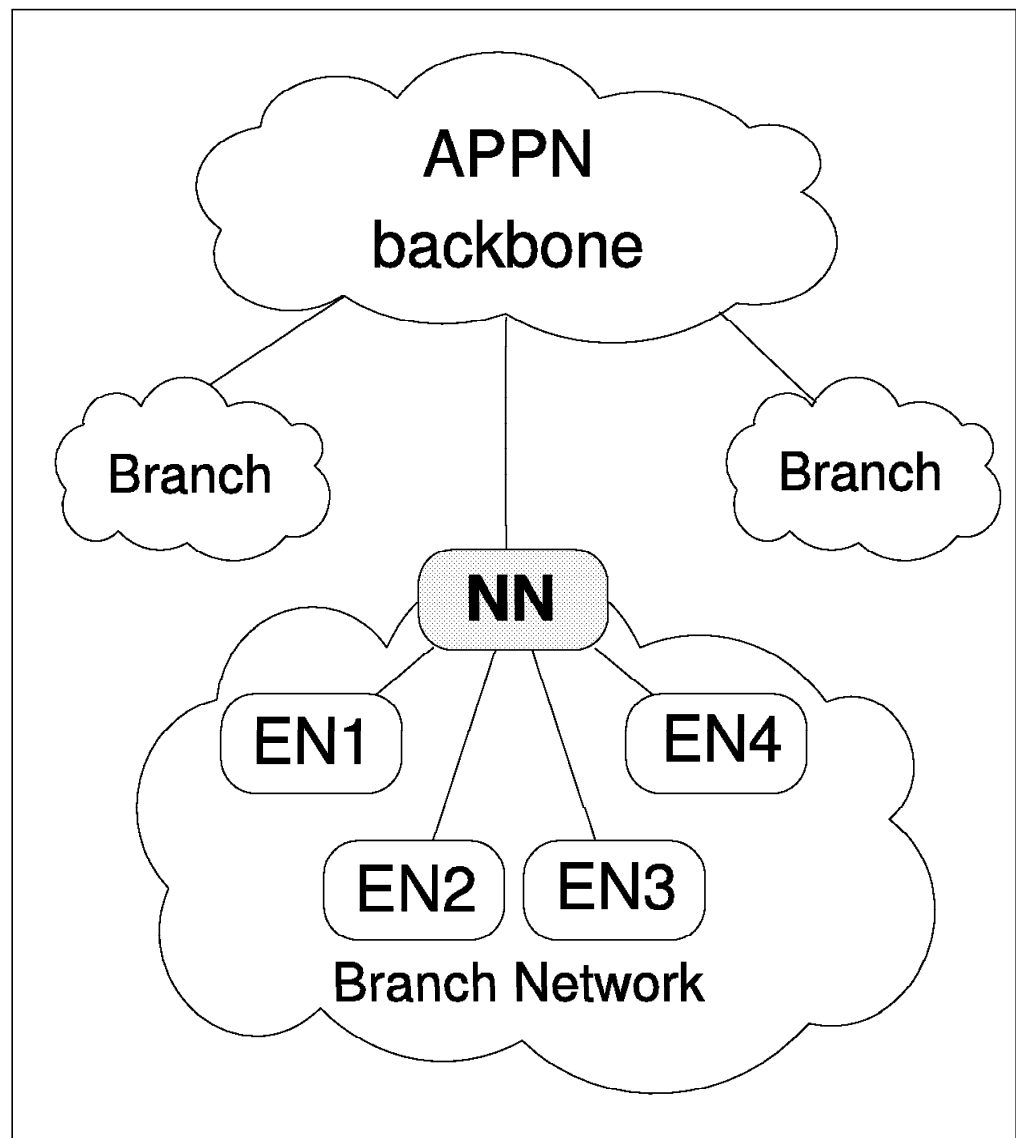


Figure 103. Traditional APPN Network

In summary, the APPN large network strategy has problems with:

- The number of NNs
- Unnecessary switched hops
- Excess CP-CP traffic
- Unintelligent searching

The reasons for these problems are described as follows:

- Topology size

Problems occur if an individual APPN topology subnetwork exceeds the size of the smallest NN's TDB storage. 3174 is the most limited NN. Its TDB limits network size to around 150 NNs. TDB size is a function of the number of NNs and links, as well as the frequency of link state changes. Many customer networks would like to install 1500-5000 small NNs as gateways to branch offices. In addition, each branch site has a LAN which is configured as a connection network, effectively doubling the number of NN TDB entries.

- Border node strategy

APPN's large network strategy has been based on border nodes, which allow you to subdivide a large APPN network into topology subnetworks of a manageable size. You need the ability to segregate APPN topology of large networks because of limited TDB storage in certain NN products. Today this is achieved using VTAM 4.2's extended border node and AS/400's peripheral border node. A gateway is necessary that can interconnect a branch office with 1-2 LANs, downstream ENs and LEN ENs with dependent and independent LUs, downstream PUs to 1-2 WAN uplinks. There's little reason for topology sharing between the WAN backbone and a branch office, or directory searches in the branch office for resources that are located elsewhere. With multiple links to the branch, searches may come in on one link and leave via another, without finding anything in the branch, aggravating network performance. PBN does not meet requirements since it only interconnects unlike NETIDs. EBN is far too large (code size) for rapid implementation on the target platforms. A new, slim, branch subnet is needed.

- Inability to exploit CDS outside the local topology subnet

Today VTAM is the only product providing an APPN Central Directory Server (CDS). Exploitation of CDS is desirable to minimize the initial directory broadcasts in separate subnets to locate a resource. When VTAM is both CDS and EBN, it caches data about resources it has located and uses it during future border node searches. However, many sessions are initiated from the mainframe site. Branch-to-branch sessions are also a problem. Which branch should be searched to find a particular LU? It is desirable to avoid even the initial broadcast to locate a peripheral resource. The problem worsens when you need to configure many branch subnets with the same NETID (a feature supported by EBN). You require this because older applications don't support NETID-qualified names. In such a network, all network directory broadcasts go to all subnets. These broadcasts could be avoided if central resource registration could be done across subnetwork boundaries (today it can't, and CDSs aren't visible outside their native topology net). When BNs are used, even if a WAN network such as X.25, ISDN, FR, or ATM allows peer-to-peer connectivity, LU-LU sessions between LUs in different APPN topology subnets must traverse at least one BN.

7.1.2 Requirements

The requirements to develop the branch extender were:

- Reduce the number of NNs in a large APPN network.
- Hide peripheral (branch) topology from the WAN.
- Allow direct, peer-to-peer communication between branches connected to the same APPN connection network.

- Reduce CP-CP session traffic on WAN links.
- Isolate the branch network from backbone WAN traffic.
- Support native and non-native branch-to-WAN interconnection.
- Minimize WAN flows.
- Support cascaded BrNNs.
- Support direct connections between two BrNNs.

Another requirement was to allow a peer BrNN to serve as a backup NNS for another BrNN (thus changing a peer configuration into a cascaded configuration), so long as the BrNN which may become cascaded is not a DLUR. Initially, one network node server per BrNN is sufficient. A BrNN must provide NN services for branch ENs, LEN ENs, dependent T2.0 and T2.1 nodes, and its own dependent PUs and LUs, if any. A BrNN must be able to host its own APPC applications. A BrNN must provide DLUR support for dependent PUs in its domain. A BrNN will register domain resources uplink to an NNS, which is expected to centrally register them at a CDS if available. It is important that a BrNN be able to use any dial backup links to support switching to a new NNS, switching CP-CP sessions, and switching session routing to an on-demand backup link.

7.2 BrNN - Design Overview

LU-LU sessions between two ENs in the same branch are managed and connected without the intervention of NNs across the WAN. However, a BrNN relies on its server for default routing to connect LU-LU sessions between LUs in different branches. A BrNN does not try to get complete topology data. It only retains the topology of its domain. When that information is insufficient, the BrNN sends a Locate to its NNS asking for assistance in locating the resource and calculating the route to get there from the BrNN.

To keep these dual faces, although it's really an NN, a BrNN emulates an EN to its NNS by setting EN parameters in XID and CP-CP Capabilities. A BrNN determines which links are uplinks by configuration.

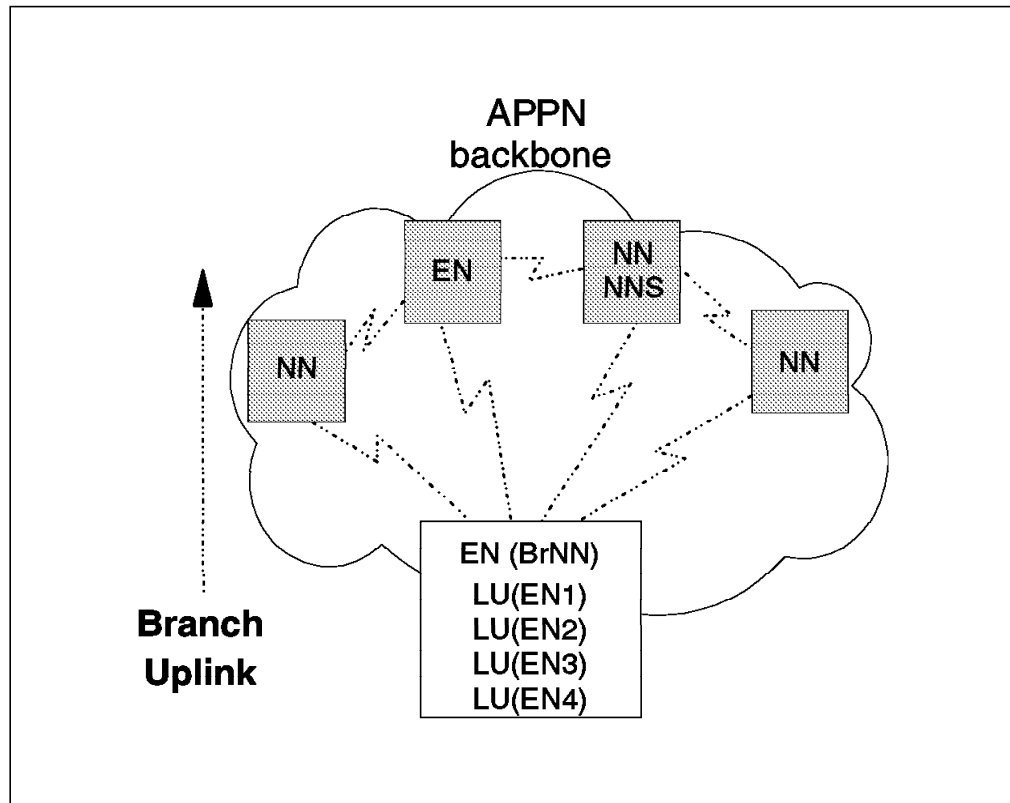


Figure 104. Branch Extender Network Node (BrNN) - Uplink Connections

A BrNN is an NN inside and shows an NN image to its domain ENs. Domain ENs don't need to know that their NNS is actually a BrNN. Before sending an XID, the BrNN determines if the link is a downlink or uplink, for example by checking the adapter or port control blocks. If it is a downlink, the BrNN sends XID3s (XID3(pn), XID3(np), XID3(na)) and later CP capabilities without modification, consistent with an NN role. XIDs indicate support for CP-CP sessions; the NNS bit is set on.

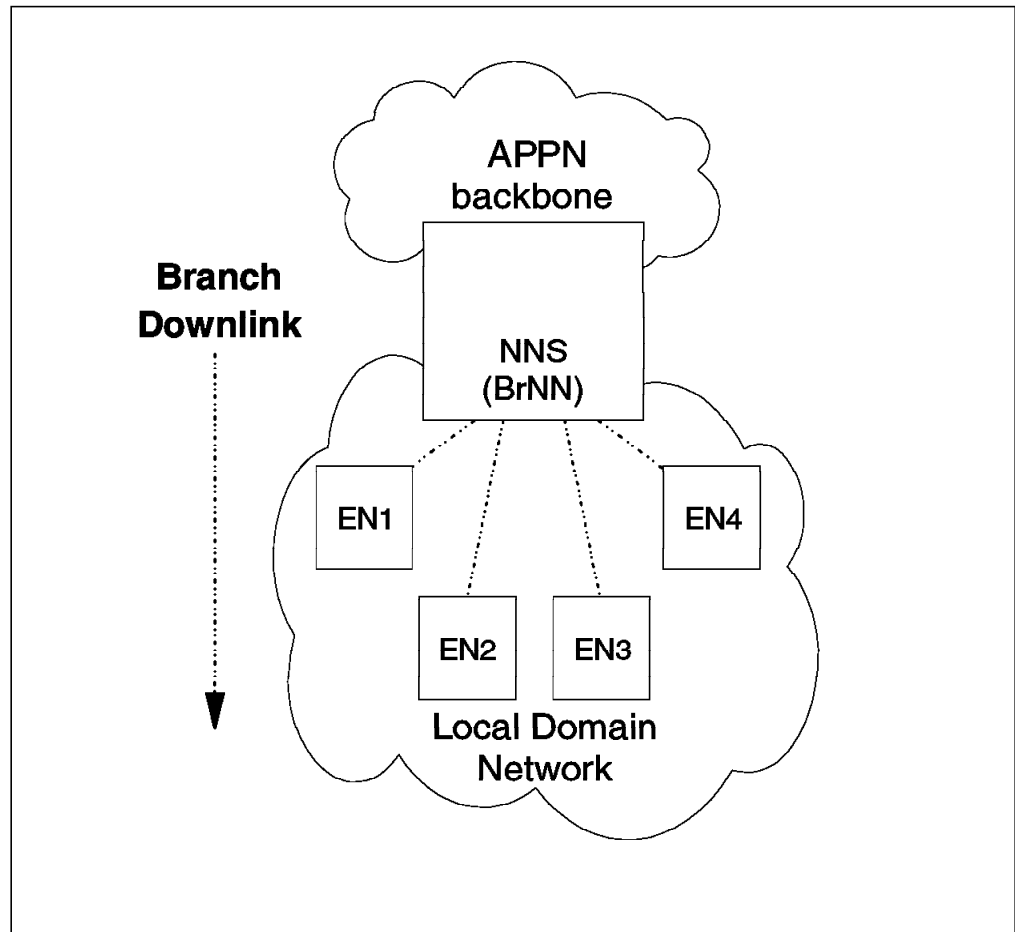


Figure 105. Branch Extender Network Node (BrNN) - Downlink Connections

Since a BrNN appears as an EN to its NNS, the BrNN's flows must not confuse the uplink network. These flows include the RSCVs in BIND, Locate, and Route Setup, the resource hierarchy in Find and Found, MDS transport, MS Capabilities, and the SNMP MIB (if supported). A BrNN will set up the information properly.

7.3 Restrictions

The following restrictions apply to networks configured to use the Branch Extender:

- A branch network node can only connect to a network node over a branch uplink.
- Illegal double agent is not allowed.
- To prevent search looping, downlink end nodes must be real end nodes, not nodes presenting an end node image.
- A node cannot have CP-CP sessions with the branch network node through both branch uplinks and branch downlinks simultaneously.
- The branch network node can have CP-CP sessions over only one branch uplink at a time; that is, it can have CP-CP sessions with at most one network node server.
- LAN DLC for PCNET does not support the branch extender feature.

- Branch extender does not provide twinax support.
- A branch extender node cannot have DLUR sessions downstream.
- More than one uplink is allowed but BrNN can have only one NNS at a time.
- Only one CP-CP session is allowed between BrNN and NNS.
- Parallel BrNNs are not supported.

If your backup link is to a Branch Extender, sessions will not switch back to the primary link if the sessions flowing over the backup link were not initiated from the local node. In this case, the switch back to the primary link will not occur until all sessions have ended. This is due to the fact that the Branch Extender is seen as an end node and it cannot update the topology database.

Configuration restrictions:

- Uplinks
 - One NNS at a time
 - Any number and type of links and nodes is OK
 - HPR and ISR are OK
 - Can switch to a new NNS if needed
- Downlinks
 - No NN, DLUR, EBN, PBN, VTAM
 - Any number and type of links is OK
 - BrNNs, ENs, LENs, T2.0 nodes OK
- Cascaded BrNNs
 - Only the hierarchically highest BrNN can be a DLUR

7.3.1 Connection Networks and BrNN

A Connection Network (CN) is a "zero-hop routing" construct in APPN that allows APPN nodes on a LAN to have direct link connection with each another. Logical link definitions are not required to each node on the LAN. Connection networks are identified by a connection network name. All machines that are participating in the connection network specify the connection network name in their configuration. The LAN itself will appear as a virtual node in the APPN network. This virtual node can be used in route selection. APPN route selection will realize that two nodes are members of the same connection network. The EN uses the LAN address to establish a direct link connection to another EN. The session flows over this direct link connection. The session does not get routed through the NN.

Note

A BrNN in eNetwork Communications Server for OS/2 Warp Version 5.0 supports connection networks over LAN (that is, token ring, ethernet) but it does not support connection networks over WAN (that is, Frame-Relay for example)

7.3.2 HPR in a BrNN

A branch extender node (BrNN) supports HPR with no restrictions. Therefore, you can configure a BrNN as:

- ANR node without RTP functions
 - RTP function resides in ENs
 - Reduces storage requirements
- RTP end point
 - There will be multiple RTP pipes for sessions with different classes of service (COS).
 - Based on your configuration options, the interactive traffic will normally have higher priority (#INTER) than batch traffic (#BATCH).
 - Additional storage is needed on BrNN to handle RTP function.

7.3.3 DLUR and BrNN

When eNetwork Communications Server for OS/2 Warp is configured as a PU 2.0 gateway with DLUR to support downstream dependent LU sessions in domain PUs, Branch Extender coexists and provides added value, for no extra cost. An internal interface is provided feeding the PU 2.0 gateway traffic into the DLUR code. Another way to view Branch Extender is as a PU 2.1 gateway for downstream ENs and LEN ENs with independent LUs, analogous to a PU 2.0 gateway for dependent LUs.

Note

In eNetwork Communications Server for OS/2 Warp Version 5.0 BrNN does not support downstream nodes with DLUR. Whether the downstream node is an end node or a cascaded branch extender node, a DLUR/DLUS pipe cannot go through a BrNN. It is expected that this limitation will be removed in a future release.

7.3.4 Cryptography and BrNN

End-to-end LU-LU cryptography for APPC and dependent LU sessions has no restrictions when using a BrNN and the sessions can flow over APPN networks, LEN connections or using AnyNet SNA/IP. However, APPN encryption support is not available in this release.

Note

APPN cryptography is not supported in a BrNN using eNetwork Communications Server for OS/2 Warp Version 5.0.

APPN cryptography is the case where the node supports APPN (CP-CP sessions) and the user selects the option to have APPN participate in the SLE (session level encryption) process by providing a CP Key Translate option. When using the CP Key Translate SLE option, the user must provide the shared key-encrypting key with another end node CP or network node CP.

CP Key Translate is an optional feature for LU 6.2 Session Level Encryption. However, when it is used, the local LU and the partner LU do not necessarily

have to share the same transport key. In this case, the session key encrypted under the partner LU key is transported using APPN commands (Locate Request and Locate Reply).

7.4 Connecting BrNNs

It is also possible to connect two or more branch extender network nodes (BrNNs). There are two ways you can do this:

1. EN to EN

Both BrNNs configure the link between them as an uplink, and appear as ENs to each other. Each BrNN typically has at least one uplink to the WAN. In this configuration the BrNN served downlink domains are disjoint, but a given EN can establish CP-CP sessions and register its resources with either BrNN. The BrNN with whom an EN currently has CP-CP sessions is its NNS; only this BrNN replies positively to a Locate request for that EN's resources.

2. Cascaded BrNNs

The upper BrNN defines the link as a downlink and assumes an NN role. The lower BrNN defines it as an uplink and assumes an EN role. The lower BrNN, in turn, may be the NNS for resources on its own downlinks.

7.5 BrNN Configuration

In this section we describe the configuration options to enable a network node server to be a branch extender network node (BrNN) and to support uplink and downlink connections.

Note

If you wish to run Branch Extender with connections to prior versions of Communications Server, you must apply APAR JR09892 to older versions.

For Communications Server, V4.0, this APAR is available in APAR fixpack CSA4017 or later. The fixpack can be downloaded from FTP site <ftp://ps.software.ibm.com/ps/products/cm2/fixes/v4server/>

For Communications Server, V4.1, this APAR is available in APAR fixpack CSA4117 or later. The fixpack can be downloaded from FTP site <ftp://ps.software.ibm.com/ps/products/cm2/fixes/v41server/>

7.5.1 Configuration Options

In this section we provide a configuration checklist to enable your network node (NN) as a branch extender node (BrNN).

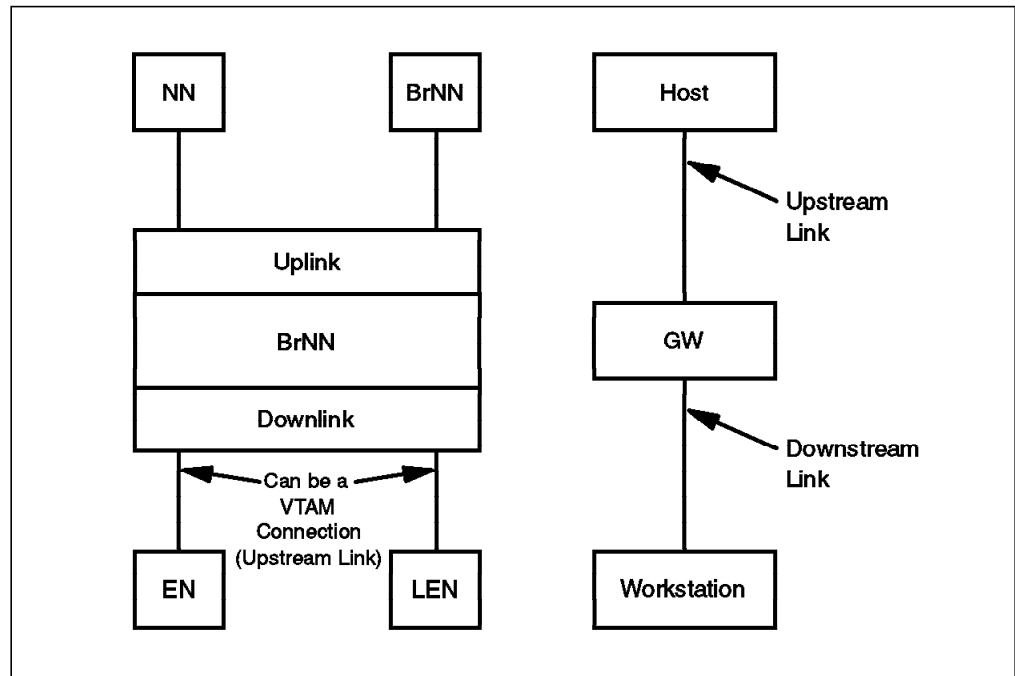


Figure 106. Branch Extender Node - Connections

The following configuration options are available:

Local node You should always enable the NN as a BrNN (see Figure 107).

Figure 107. Branch Extender Configuration - Local Node

Uplink connections

You will always enable the uplink connections to support BrNN, for example, to the network node server (NNS) or to another BrNN if using cascaded BrNNs. This can be accomplished in two different ways:

1. DLC profile. You may want to include BrNN support for all you uplink connections using this device. The uplink connections can be either explicitly defined (define logical link) or they can be implicit connections (see Figure 108). For explicit uplink connections you can always override the option in the logical link definition.

Note

For implicit uplinks, you must always set the BrNN option in the DLC profile.

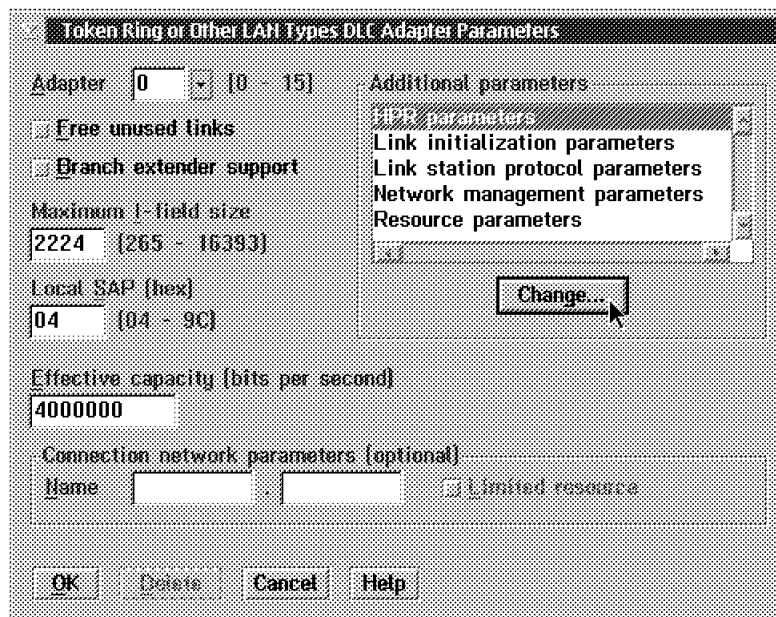


Figure 108. Branch Extender Option - DLC Profile

2. Logical link or connection profile. If you did not configure BrNN support in the DLC profile, you will use the override option, in the logical link definition, to enable the explicit uplink for BrNN support (see Figure 109 on page 169).

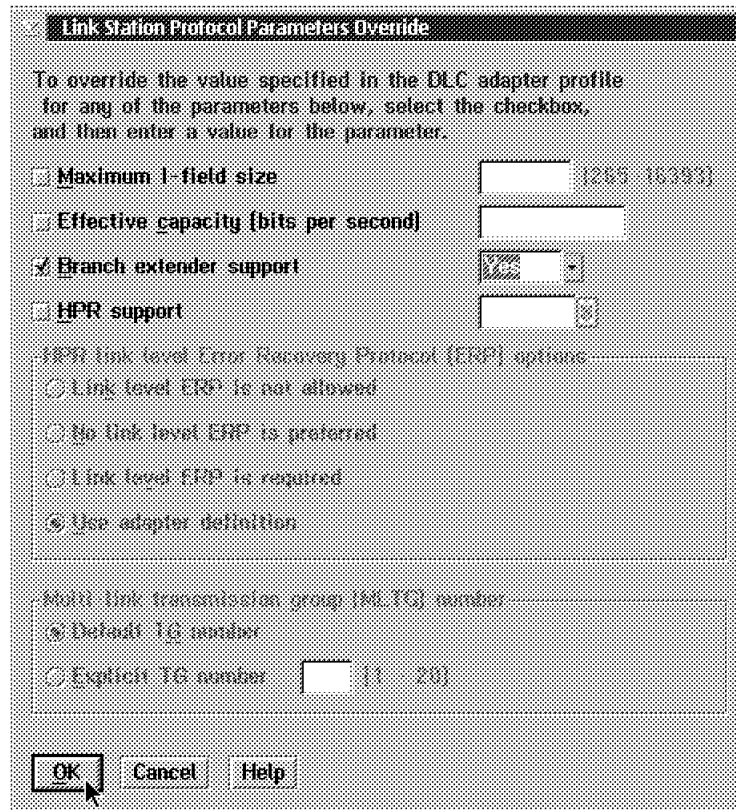


Figure 109. BrNN Uplink Connection - Override Option

Downlink connections

You never enable BrNN for downlink connections.

If the local node supports Branch Extender uplinks, all links to network nodes must be Branch Extender uplinks. This is needed to make sure the node is treated as an end node by adjacent network nodes.

If branch network nodes have links defined between one another, they must be defined as peer connections, which give them a link type of LEARN in the response file. Alternatively, you can define the link as Active at startup equal to Yes so that the link is always active. When branch nodes connect to each other, loops in the topology occur. This is acceptable as long as the links are always active or the links are learned as they are activated. If links are defined between branch network nodes as "end node" or "network node" links, then the topology reported upstream might interpret the links inappropriately and cause allocation failures.

7.5.2 Response File - BrNN Configuration

The following options are set when configuring a network node as a branch extender node (BrNN):

```
LOCAL_CP = (
    NAME = USIBMRA.SERVERA
    :
    BRANCH_EXTENDER_SUPPORT = 1
    :
)
```

```

LAN_DLC = (
:
  BRANCH_EXTENDER_SUPPORT = 0
:
)
SDLC_DLC = (
:
  BRANCH_EXTENDER_SUPPORT = 0
:
)
LOGICAL_LINK = (
  NAME = SRVA2RAK
  DLC_NAME = IBMTRNET
  :
  BRANCH_EXTENDER_UPLINK = 1
  :
)

```

7.5.3 NDF File - BrNN Configuration

The local CP and the logical link profiles are included in the NDF configuration. These options can be changed directly using an ascii editor but the configuration will have to be verified using the cmverify command.

Note

The DLC profile is not included in the NDF configuration file.

The following options are set when configuring a network node as a branch extender node (BrNN):

```

DEFINE_LOCAL_CP  FQ_CP_NAME(USIBMRA.SERVERA  )
                  CP_ALIAS(SERVERA  )
                  NODE_TYPE(NN)
                  :
                  BRANCH_EXTENDER_SUPPORT(YES)
                  :
DEFINE_LOGICAL_LINK  LINK_NAME(SRVA2RAK)
                     ADJACENT_NODE_TYPE(NN)
                     :
                     BRANCH_EXTENDER_UPLINK(YES)
                     :

```

Chapter 8. AnyNet SNA over TCP/IP - TCP62 API

IBM TCP62 is an Application Programming Interface (API) that simplifies configuration of AnyNet LU 6.2 support over TCP/IP. In this chapter we illustrate the required steps you need to do in order to easily implement SNA over TCP/IP support for LU 6.2 sessions.

8.1 Overview

The TCP62 configuration can be thought of as a filter on top of the existing Communications Server configuration API. It adds function by adding new and extended verbs that are useful in meeting TCP62 requirements. In particular, since many configuration files are identical except for the local LU names, unique local LU names can be defined dynamically based on the local IP address. Since TCP62 always implies LU 6.2 over IP, SNA and AnyNet configurations can be simplified to one or two parameters.

TCP62 consists of the following four verbs:

START_TN62

Builds a Communications Server configuration file based on the parameters passed in the START_TN62 verb and starts Communications Server using that configuration file.

DEFINE_PARTNER_LU_TN62

Generates a partner LU name and passes the define verb to the Communications Server node

DEFINE_LOCAL_LU_TN62

Generates a local LU name and passes the define verb to the Communications Server node.

STOP_TN62

Stops Communications Server and immediately ends any communication in progress.

The START_TN62 verb can be used when another subsystem (for example, a CICS client) wants to enable LU 6.2 over IP communication. Since START_TN62 may involve starting the underlying communications node, it should be used infrequently due to the potentially long processing time. For example, START_TN62 can be used when the subsystem is initialized and should not be used on a per transaction basis. Similarly, it is expected that both the DEFINE_LOCAL_LU_TN62 and DEFINE_PARTNER_LU_TN62 verbs would only be used during subsystem initialization. However, the processing time for these two verbs is much less than for the START_TN62 verb.

8.1.1 Writing TCP62 Programs

Communications Server TCP62 provides a dynamic link library (DLL) file that handles TCP62 verbs.

TCP62 verbs have a straightforward language interface. Your program fills in fields in a block of memory called a verb control block (VCB). Then it calls the TCP62 DLL and passes a pointer to the VCB. When the program is complete, TCP62 returns, having used and then modified the fields in the VCB. Your program can then read the returned parameters from the VCB.

The following table shows source module usage of supplied header files and libraries needed to compile and link TCP62 programs. Some of the header files may include other required header files.

Header File and Library for TCP62:

Operating System	Header File	Library	DLL Name
OS/2	TN62API.H	ACS32.LIB	TN62API.DLL

8.1.2 TN62API() Entry Point

This provides a synchronous entry point for issuing the following TCP62 API verbs:

- START_TN62
- STOP_TN62
- DEFINE_LOCAL_LU_TN62
- DEFINE_PARTNER_LU_TN62

Syntax

- void APIENTRY TN62API(PVOID vcb);

Parameter	Description
vcb	Pointer to verb control block

Returns

No return value. The primary_rc and secondary_rc fields in the verb control block indicate any error.

Remarks

This is the main synchronous entry point for the TCP62 services API. This call blocks until the verb completes.

8.1.3 Migration Considerations

One common migration scenario involves the Communications Server node communicating outside the scope of TCP62. For example, a user may have previously installed Communications Server and used it for emulator or native SNA communication. In this case, the user will have a default Communications Server response file and Communications Server may be running when

START_TN62 is invoked. This section discusses the design and use of the TCP62 API using this scenario.

When START_TN62 creates its response file, it takes as its starting point the active or default response file. Therefore, any Communications Server configuration, such as links, modes, or LUs performed outside of TCP62 is not lost.

AnyNet support, (that is, whether or not a node can use AnyNet IP transport), cannot be dynamically changed in a running node. For example, if a node is running that is not configured to support AnyNet, the START_TN62 verb cannot dynamically update the running node to support AnyNet and START_TN62 will fail. In this case, Communications Server must be stopped before START_TN62 can complete successfully.

The following node parameters cannot or should not be changed in a running node:

- CP name
- CP alias
- SNA domain name suffix
- AnyNet timer values

If the node is running and START_TN62 is issued with parameters from the above list that are different from those in the running node, the START_TN62 completes successfully. However, the values from the running node will be unchanged and these values will be returned in the START_TN62 VCB. The START_TN62 tells the caller that some values were not used by setting primary_rc to 0 and secondary_rc to TN62_PARAMETERS_NOT_USED.

8.1.4 Dynamic Name Generation

TCP62 dynamically generates local LU or partner LU names if the input name parameter is a template. That is, if it contains one or more replacement characters `"*"`. The name generation algorithm is also used by the SXMAP program in Communications Server.

The following example shows how this algorithm is implemented.

```

static void
SxMap (unsigned char *generatedName,
       unsigned char *nameTemplate,
       unsigned int templateLength,
       unsigned long addr,
       unsigned long mask)
{
    int            I;
    unsigned long  host_bits;
    unsigned long  bit_pos;
    char chars[]   = "0123456789ABCDGHIJKLMNOPQRSTUVWXYZEI@#$%&";
    addr           = ntohl(addr);
    mask           = ntohl(~mask);
    host_bits      = 0L;
    bit_pos        = 0x00000001;

    for (i=0; i<32; i++)
    {
        if (mask & bit_pos)
        {
            host_bits |= (addr & bit_pos);
            bit_pos <<= 1;
        }
        else
        {
            addr >>= 1;
            mask >>= 1;
        }
    }

    for (i = templateLength; i >= 0; i--)
    {
        if (nameTemplate[i] == REPLACEMENT_CHAR)
        {
            generatedName[i] = chars[host_bits & 0x1F];
            host_bits >>= 5;
        }
        else
            generatedName[i] = nameTemplate[i];
    }
    return;
}

```

Figure 110. Algorithm to Dynamically Generate Local LU Name Based on IP Address

The algorithm selects bits from `addr`, which is a local or remote IP address. The selected bits are those where the corresponding bit in the mask is 0. The selected bits are then taken in groups of 5, right to left, to generate a character for each replacement character in `nameTemplate`.

For example, if `nameTemplate = "A*NAME*"`, `addr = 0x13.0x8f.0x22.0xa3` and `mask = 0xff.0xff.0xff.0x00`. The bits selected by mask are `0x00.0x00.0x00.0xa3`. Since there are two replacement characters in `nameTemplate`, the two groups of five bits are `0x05` and `0x03`. Using these as indices in `chars` yields a `generatedName` of `"A2NAME4"`.

8.2 TCP62 API Support

Communications Server supports the following verbs using the TCP62 API.

8.2.1 START_TN62

The START_TN62 verb starts the Communications Server node.

The structure of this verb is shown below.

```
typedef struct start_tn62
{
    unsigned short    opcode;        /* verb operation code */
    unsigned char     reserv1[6];    /* reserved */
    unsigned short    primary_rc;    /* primary return code */
    unsigned char     reserv2[2];    /* reserved */
    unsigned long     secondary_rc;  /* secondary return code */
    unsigned char     reserv3[4];    /* reserved */
    unsigned char     key[8];        /* key (ASCII) */
    unsigned char     fqcp_name[17]; /* real fully-qualified */
                                   /* name or a template */
                                   /* for a fully-qualified */
                                   /* name */
    unsigned char     cp_alias[8];   /* ASCII CP alias */
    unsigned char     reserv4[3];    /* reserved */
    unsigned long     ip_address_mask; /* mask used in dynamic */
                                   /* CP name generation */
    unsigned short    connection_retry_secs;
                                   /* connection retry count */
    unsigned short    unacked_dg_retry_secs;
                                   /* unacknowledged data- */
                                   /* gram retry interval */
    unsigned short    unsent_dg_retry_secs;
                                   /* unsent datagram */
                                   /* retry interval */
    unsigned short    inactivity_timer_secs;
                                   /* remote node inactiv- */
                                   /* ity poll interval */
    unsigned short    connwait_secs; /* connection wait time */
                                   /* limit */
    unsigned char     domain_name_suffix[220];
                                   /* domain name suffix */
} START_TN62;
```

Figure 111. Verb Control Block (VCB) Structure

8.2.2 Supplied Parameters

The application supplies the following parameters:

opcode	AP_START_TN62.
key	Specifies either the master or service key if the keylock feature has been activated. This is an 8-byte ASCII character string. If the name is less than 8 bytes, it must be padded on the right with ASCII blanks.

fqcp_name	This is either a real fully-qualified CP name or a template for a fully-qualified CP name. If there are no template replacement characters "*", it is a real name; otherwise, it is a template. The net ID must not contain any template replacement characters and the CP name
cp_alias	This is the ASCII CP alias for the TCP62 node. If the node is running when the START_TN62 is issued, this will contain the CP alias of the running node on return. If this field is all blanks or nulls, and the node is not running when the START_TN62 is issued, the CP alias is set to the (unqualified) CP name on return.
ip_address_mask	This is the mask to be used in dynamic CP name generation. It is ignored if fqcp_name is not a template. The mask is encoded as a big-Endian long, that is, high-order byte first to low-order byte last.
connection_retry_secs	The connection retry count is the maximum time, in seconds, for LU 6.2 over TCP/IP to set up a multiprotocol transport network (MPTN) connection over TCP/IP. When an MPTN connection setup fails, Communications Server tries every IP address associated with an LU name in the domain name server or HOSTS file until all the addresses are exhausted or until the time specified is reached. Specify a value between 1 and 65535 seconds. Default: 300 If you are unsure about what value to enter, use the default.
unacked_dg_retry_secs	The unacknowledged datagram retry interval is the maximum time, in seconds, that LU 6.2 over TCP/IP waits to resend an unacknowledged out-of-band (OOB) or MPTN keepalive datagram. When expedited data is sent over TCP/IP, this interval is used to help control the delivery of expedited data in congested situations. In SNA, some control messages are sent over TCP/IP, this interval is used to help control the delivery of expedited data in congested situations. In SNA, some control messages are sent as expedited data (for example, messages requesting the right to send data or messages taking down a session). Expedited data is not subject to congestion control and can move ahead of normal, non-expedited data. To assure delivery, AnyNet SNA over TCP/IP might send expedited data as normal data and as an OOB datagram. Specify a value between 1 and 65535 seconds. Default: 10 If you are unsure about what value to enter, use the default.

unsent_dg_retry_secs

The unsent datagram retry interval is the maximum time, in seconds, that Communications Server waits for an acknowledgment after sending expedited data on a TCP connection, before sending the data as an out-of-band (OOB) datagram.

When expedited data is sent over TCP/IP, this interval is used to help improve the delivery of expedited data in congested situations. In SNA, some control messages are sent as expedited data (for example, messages requesting the right to send data or messages taking down a session). Expedited data is not subject to congestion control and can move ahead of normal, non-expedited data. To assure delivery, AnyNet SNA over TCP/IP might send expedited data as normal data and as an OOB datagram.

Specify a value between 1 and 65535 seconds.

Default: 3

If you are unsure about what value to enter, use the default.

inactivity_timer_secs

The remote node inactivity poll interval is the number of seconds of inactivity allowed between two partner nodes before LU 6.2 over TCP/IP tries to determine whether the partner node is still active.

Type a value between 1 and 65535 seconds.

Default: 30

Setting the interval below 10 seconds might seriously affect system performance.

To calculate how long it takes before an inactive partner is detected:

1. Multiply the value of the unsent datagram retry interval by 5.
2. Add the remote node inactivity poll interval value. The resulting value is the number of seconds it takes to detect an inactive partner.

If you are unsure about what value to enter, use the default.

connwait_secs

The connection wait time limit is the maximum time, in seconds, that LU 6.2 over TCP/IP waits to receive a multiprotocol transport network (MPTN) connection or connection response packet after the TCP connection is established. This limit prevents the connecting node from waiting too long for a session partner to send a packet.

Specify a value between 1 and 65535 seconds.

Default: 30

If you are unsure about what value to enter, use the default.

domain_name_suffix The SNA domain name suffix is used when a domain name is created from the fully-qualified partner LU name.

The SNA domain name suffix is a user-defined domain name suffix created using the hierarchical naming format recognized by TCP/IP. For example, SNA.IBM.COM is an SNA domain name suffix.

Consult your network administrator to obtain an SNA domain name suffix. The suffix consists of strings concatenated with periods. Each string must be less than or equal to 63 characters, with the total length less than or equal to 237 characters.

Valid characters for each string are:

The first character must be an alphabetic character (A-Z, a-z).

The last character must be an alphanumeric character (A-Z, a-z, 0-9).

The remaining characters can be alphanumeric characters (A-Z, a-z, 0-9) or the special character (-).

Default: SNA.IBM.COM

8.2.3 Returned Parameters

See 8.2.13, "TCP62 Return Codes" on page 185 for information on primary and secondary return codes.

primary_rc If the verb succeeds, the primary return code is always zero.

secondary_rc The secondary return code qualifies the primary return code.

fqcp_name This is the real fully-qualified CP name. If the supplied fqcp_name was a template, it is replaced with the generated name.

8.2.4 STOP_TN62

The STOP_TN62 verb stops the Communications Server node. Any communications that are in progress will end.

The structure of this verb is shown below.

```
typedef struct stop_tn62
{
    unsigned short    opcode;        /* verb operation code */
    unsigned char     reserv1[6];    /* reserved */
    unsigned short    primary_rc;    /* primary return code */
    unsigned char     reserv2[2];    /* reserved */
    unsigned long     secondary_rc;  /* secondary return code */
    unsigned char     reserv3[4];    /* reserved */
    unsigned char     key[8];        /* key (ASCII) */
} STOP_TN62;
```

Figure 112. Verb Control Block (VCB) Structure

8.2.5 Supplied Parameters

The application supplies the following parameters:

opcode	AP_STOP_TN62.
key	Specifies either the master or service key if the keylock feature has been activated. This is an 8-byte ASCII character string. If the name is less than 8 bytes, it must be padded on the right with ASCII blanks.

8.2.6 Returned Parameters

See 8.2.13, "TCP62 Return Codes" on page 185 for information on primary and secondary return codes.

primary_rc	If the verb succeeds, the primary return code is always zero.
secondary_rc	The secondary return code qualifies the primary return code.

8.2.7 DEFINE_LOCAL_LU_TN62

TCP62 extends the DEFINE_LOCAL_LU verb to allow definitions of LU names that are generated dynamically, based on a supplied template, mask, and the local IP address.

The structure of this verb is shown below.

```

typedef struct define_local_lu_tn62
{
    unsigned short opcode;           /* verb operation code */
    unsigned char reserv1[6];        /* reserved */
    unsigned short primary_rc;       /* primary return code */
    unsigned char reserv2[2];        /* reserved */
    unsigned long secondary_rc;      /* secondary return code*/
    unsigned char reserv3[4];        /* reserved */
    unsigned char key[8];            /* key (ASCII) */
    unsigned char lu_name[8];        /* local Lu name */
    unsigned char lu_alias[8];       /* Lu alias (ASCII) */
    unsigned char nau_address;        /* network addressable */
                                      /* unit address */

    unsigned char external_support;  /*external support for sync point */
                                      /*AP_NONE x'00' */
                                      /*AP_SYNCPT_PROVIDER x'80' */
                                      /*AP_REMOTE_TP_PROVIDER x'40' */
                                      /*AP_SYNCPT_AND_REMOTE_TP x'CO' */

    unsigned char host_link_name[8]; /* host link name */
                                      /* 0 or 1-8 bytes */
                                      /* (EBCDIC type A) */

    unsigned char lu_model_name[7];  /* self-defining dep LU */
                                      /* model name */

    unsigned char reserv4[35];       /* reserved */
    unsigned long ip_address_mask;   /* mask used in CP name */
    unsigned char reserv5;           /* reserved */
} DEFINE_LOCAL_LU_TN62;

```

Figure 113. Define Local LU Verb Control Block Structure

8.2.8 Supplied Parameters

The application supplies the following parameters:

opcode	AP_DEFINE_LOCAL_LU_TN62.
key	Specifies either the master or service key if the keylock feature has been activated. This is an 8-byte ASCII character string. If the name is less than 8 bytes, it must be padded on the right with ASCII blanks.
lu_name	This is either a real LU name or an LU name template. If there are no template replacement characters "*", it is a real name; otherwise it is a template. Except for replacement characters, lu_name must be a valid, EBCDIC LU name. The first character must not be the replacement character.
lu_alias	The 8-byte ASCII name used locally for the LU. The name is not sent outside the local node.
nau_address	Specifies the network addressable unit (NAU) address of the LU.

You can specify a value from 0 to 254, where:

0 Specifies that the NAU address is not used, and the LU is an independent LU. Sessions among APPN end nodes and network nodes must use independent LUs. An LU type 6.2 is the only type of SNA LU that supports independent sessions. An independent session does not depend on an SSCP (that is, the LU can send a BIND directly without the help of an SSCP).

1-254 Specifies the NAU address of the LU, and that the LU is a dependent LU for sessions to a subarea node. If your network contains a subarea that does not support an independent session from a peripheral node, you will be restricted to a single dependent session between your APPC LU and that subarea. In this case, the LU will need to be assigned a unique NAU address.

An LU's NAU address is the address used by a subarea node for an LU dependent session. A dependent session is a session that depends on an SSCP to initiate it (that is, an optional INIT_SELF request that flows on the LU-to-SSCP session, the SSCP sends a CINIT request to a subarea LU, and the subarea LU sends the BIND).

On a given link, every LU that uses a dependent session must be assigned a unique NAU address. Every LU defined for an LUA session, every LU defined for a 3270 session, and every LU defined for a 3270 gateway session, uses a dependent session and must be assigned a unique NAU address.

host_link_name

The 8-byte EBCDIC name of the local link station. This logical link can be activated by specifying this name on the ACTIVATE_LOGICAL_LINKS verb.

Note: The host link name is ignored if NAU_address equals 0 is specified. Otherwise, this parameter is optional. If a NAU address is specified, the LU definition is assigned to the CP-PU.

This is a Type A EBCDIC character string. If the name is less than 8 bytes, it must be padded on the right with EBCDIC blanks. In addition, the string cannot begin with an EBCDIC character "@" (0x7c).

LU_model_name	This is the 7-byte EBCDIC model name of the dependent LU 6.2 local LU. This is a 7-byte EBCDIC character string consisting of uppercase A-Z and 0-9 only. If the lu_model_name is not used, it must be filled with EBCDIC blanks. It should match a model name on a Host Model definition.
ip_address_mask	This is the mask to be used in dynamic LU name generation. It is ignored if lu_name is not a template. The mask is encoded as a big-Endian long, that is, high-order byte first to low-order byte last.

8.2.9 Returned Parameters

See 8.2.13, “TCP62 Return Codes” on page 185 for information on primary and secondary return codes.

primary_rc	If the verb succeeds, the primary return code is always zero.
secondary_rc	The secondary return code qualifies the primary return code.
lu_name	This is the real name of the defined LU. If the supplied lu_name was a template, it is replaced with the generated name.
lu_alias	If all blank or nulls on input, this will contain the LU name (in ASCII) on output.

8.2.10 DEFINE_PARTNER_LU_TN62

TCP62 extends the DEFINE_PARTNER_LU verb to allow the definition of LU names that are generated dynamically, based on a supplied template, mask, and IP address. The verb is extended by using a new overlay structure. The DEFINE_PARTNER_LU_TN62_OVERLAY is specified after all the alt_alias_overlay structures (if any) in the DEFINE_PARTNER_LU VCB.

The opcode field in the DEFINE_PARTNER_LU VCB must be set to AP_DEFINE_PARTNER_LU_TN62 before invoking the TCP62 API. For a description of a DEFINE_PARTNER_LU verb, see *Communications Server for OS/2 Warp: System Management Programming Reference*.

The structure of this verb is shown below.

```
typedef struct define_partner_lu_tn62_overlay
{
    unsigned long ip_address_mask;          /* mask used in dynamic */
                                           /* name generation */
    unsigned long partner_ip_addr;          /* IP address of the */
                                           /* partner LU */
    unsigned char partner_hostname[220];    /* host name of the */
                                           /* partner LU */
} DEFINE_PARTNER_LU_TN62_OVERLAY;
```

Figure 114. Define Partner LU Verb Control Block Structure

8.2.11 Supplied Parameters

The application supplies the following parameters:

opcode	AP_DEFINE_PARTNER_LU_TN62.
fq_partner_lu_name	This is either a real fully-qualified partner LU name or a template for a fully-qualified partner LU name. If there are no template replacement characters "*", it is a real name; otherwise it is a template. The net ID must not contain any template replacement characters. Note that the replacement character is "*" instead of "." to avoid confusion with the "." separating the net ID and partner LU name.
ip_address_mask	This is the mask to be used in dynamic partner LU name generation. It is ignored if fqplu_name is not a template. The mask is encoded as a big-Endian long, that is, high-order byte first to low-order byte last. If the fqplu_name is a template, one of the following two parameters must be specified. If both are specified, partner_ip_addr takes precedence.
partner_ip_addr	This is the IP address of the partner LU. It is ignored if fqplu_name is not a template. All zeros indicate "unspecified". The IP address is encoded as a big-Endian long, that is, high-order byte first to low-order byte last.
partner_hostname	This is the host name of the partner LU. It is ignored if the fqplu_name is not a template. All blanks indicate "unspecified".

8.2.12 Returned Parameters

See 8.2.13, "TCP62 Return Codes" for information on primary and secondary return codes.

primary_rc	If the verb succeeds, the primary return code is always zero.
secondary_rc	The secondary return code qualifies the primary return code.
fq_partner_lu_name	This is the real fully-qualified LU name. If the supplied fq_partner_lu_name was a template, it is replaced with the generated name.

8.2.13 TCP62 Return Codes

The following section summarizes the unique TCP62 return codes. Note, DEFINE_LOCAL_LU_TN62 and DEFINE_PARTNER_LU_TN62 may also have return codes that are described in *Communications Server for OS/2 Warp: System Management Programming Reference*. Each subsection heading lists both the primary and secondary return codes in parentheses (primary_rc, secondary_rc), using defined symbols located in tn62api.h or appcdef.h.

(TN62_ERROR, TN62_NODE_RUNNING_NO_ANYNET)

returned by	START_TN62
--------------------	------------

cause The Communications Server node is running and the running configuration does not support AnyNet. START_TN62 did not complete successfully.

corrective action Stop the node and reissue the START_TN62.

(TN62_ERROR, TN62_CONFIGURATION_FILE_ERROR)

returned by START_TN62

cause A TCP62 call to the Communications Server configuration API failed. The most likely cause is a TCP62 or Communications Server program defect. START_TN62 did not complete successfully.

corrective action None. Collect problem determination data by turning on all tracing and re-creating the problem. Also, capture any log data.

(TN62_ERROR, TN62_NODE_NOT_STARTED)

returned by START_TN62

cause The node failed to start. The most likely cause is a TCP62 program defect. START_TN62 did not complete successfully.

corrective action None. Collect problem determination data by turning on all tracing and re-creating the problem. Also, capture any log data.

(TN62_ERROR, TN62_NODE_START_INCOMPLETE)

returned by START_TN62

cause The node started, but the configuration was not successful. The most likely cause is the AnyNet program is not installed. The node is running.

corrective action Ensure the AnyNet component is installed. If the problem persists, collect problem determination data by turning on all tracing and recreating the problem. Also, capture any log data.

(AP_OK, TN62_PARAMETERS_NOT_USED)

returned by START_TN62

cause The AnyNet parameters (timers and domain_name_suffix) or both the CP name and CP alias from START_TN62 were not used. The node was already running using different parameters. The parameter values used by the running node are returned in the START_TN62 VCB. START_TN62 completes successfully.

corrective action None.

(TN62_ERROR, TN62_NAME_GENERATION_ERROR)

returned by START_TN62, DEFINE_LOCAL_LU_TN62,
DEFINE_PARTNER_LU_TN62

cause	Dynamic name generation failed due to gethostname (for START_TN62 and DEFINE_LOCAL_LU_TN62) or gethostbyname (for DEFINE_PARTNER_LU_TN62). The most likely causes are TCP/IP was not installed, configured and active, or an incorrect partner_hostname passed on DEFINE_PARTNER_LU_TN62. The verb does not complete successfully.
corrective action	Ensure that TCP/IP is configured and active, and that the partner_hostname on DEFINE_PARTNER_LU_TN62 is correct.
(AP_PARAMETER_CHECK, INVALID_CP_NAME)	
returned by	START_TN62
cause	The fqcp_name is not valid. The net ID must not contain any template replacement characters and the CP name must not begin with a replacement character. Except for the replacement character, this must be a valid EBCDIC fully-qualified CP name. START_TN62 does not complete successfully.
corrective action	Correct the fqcp_name parameter.
(AP_PARAMETER_CHECK, INVALID_LU_NAME)	
returned by	DEFINE_LOCAL_LU_TN62
cause	The lu_name is not valid. The LU name must not begin with a replacement character. Except for the replacement character, this must be a valid EBCDIC LU name. DEFINE_LOCAL_LU_TN62 does not complete successfully.
corrective action	Correct the lu_name parameter.
(AP_PARAMETER_CHECK, INVALID_FQ_LU_NAME)	
returned by	DEFINE_PARTNER_LU_TN62
cause	The fqplu_name is not valid. The net ID must not contain any template replacement characters and the partner LU name must not begin with a replacement character. Except for the replacement character, this must be a valid EBCDIC fully-qualified LU name. This return code will also occur if the fqplu_name is a valid template, but both partner_ip_addr and partner_hostname are unspecified. DEFINE_PARTNER_LU_TN62 fails.
corrective action	Correct the fqplu_name parameter or the (partner_ip_addr, partner_hostname) combination.

8.3 TCP62 Configuration

In order to explain the TCP62 support feature configuration in IBM Communication Server Release 5.0 for OS/2 Warp, we set up a scenario that consists of one workstation running eNetwork Communications Server for OS/2 Warp Version 5.0 and another running the OS/2 Access Feature connected together via a token-ring network. Each computer was running the TCP/IP protocol and was able to PING the other workstation.

Note: To test this API you may require a sample test program. You may decide to write your own program or obtain one ready made. In this chapter we use the sample program **TN62TEST.EXE**. To obtain a test sample program for TCP62 support, please contact your local IBM representative in your country. For a source listing of sample program TN62TEST see Appendix A, "TCP62 API Sample Program" on page 267.

The recommended sequence for TCP62 installation and configuration is:

- OS/2 Command prompt / Command Line prompt
 1. Make a HOSTS file in the:
C:\MPTN\ETC\

directory. (This file contains mappings of TCP/IP addresses to SNA addresses.)
- CMSETUP:
 1. Install the 32-bit toolkit for the TCP62 support (TN62API.H, ACS32.LIB) in CMSETUP, if you want to write your own TCP62 programs using the TCP62 DLL.
 2. Make an Initial (DUMMY) configuration in eNetwork Communications Server for OS/2 Warp Version 5.0 for SNA Local Node Parameters only for both workstations that you want TCP62 support for. (No other configuration should be made.) Note that this should only be done if you don't already have a valid configuration set up.
- OS/2 Command prompt / Command Line prompt:
 1. Copy the TN62TEST.EXE program file to the C:\CMLIB\ directory.
 2. RUN the TN62TEST.EXE program with the START verb and the fully qualified CP name as parameters.
 3. APING to the other workstation.

8.3.1 HOSTS Files

You can either use a Domain Name Server (DNS) or a hosts file for your SNA/IP address mappings. In our sample installation we used a hosts file.

The main advantage of using a DNS, instead of a hosts file, is that a DNS allows you to use wildcards for multiple definitions. Using wildcards simplifies your address mapping process.

The following are samples of hosts files that we used. They contain address mappings from TCP/IP address formats to SNA address formats.

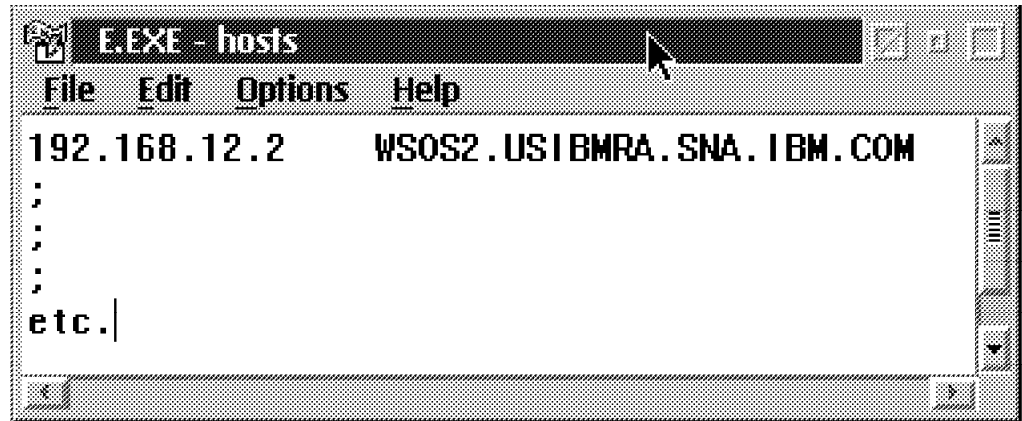


Figure 115. Host Table on Server B

The hosts files should be in the \mptn\etc subdirectory in your computer.

Note

If you are not sure about the path for the hosts file, display the etc environment variable by entering the set etc command.

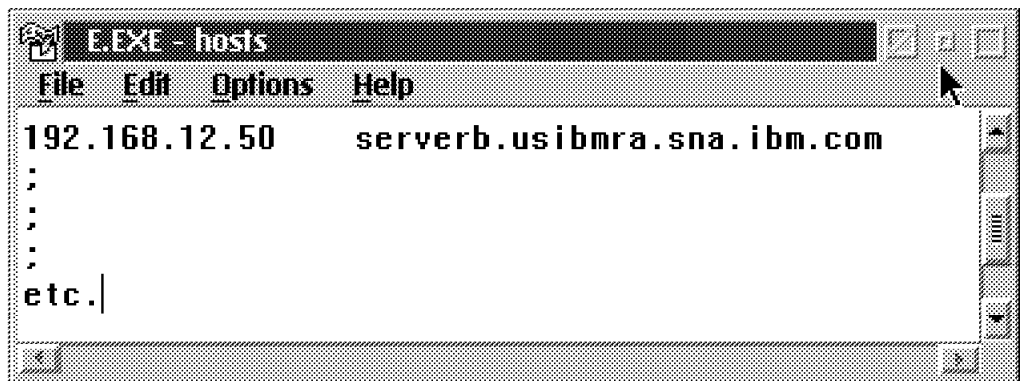


Figure 116. Host Table on OS/2 Workstation

8.3.2 SNA Local Node Configuration in Server B and OS/2 Workstation

Local Node Characteristics

Network ID: USIBMRA

Local node name: SERVERB

Node type:

- ☐ End node
- ☒ Network node
- ☒ Branch extender support

Local node ID: [hex] 05D 00001

Local node alias name: serverb

Maximum compression level: NONE

Maximum compression tokens: 0 (0 - 30400)

☒ Activate Attach Manager at start up

☐ Search required

Optional comment:

OK NetWare(R)... Cancel Help

Figure 117. SNA Local Node Parameters on Server B

1. If you are updating a configuration, you will need to provide a minimum configuration.
2. Enter the network ID and local node name for the server. The local node name must be unique in your network.
3. Enter the local node ID if required by your network.
4. Enter the local node alias name if required. This alias is used by applications not using the CP-LU fully-qualified name.

Node type		End node
Network ID	USIBMRA	
Local node name	WKS0S2	
Local node ID	(hex) 05D	00000
Local node alias name	wksos2	
Maximum compression level	NONE ▾	
Maximum compression tokens	0	(0 - 30400)
☒ Activate Attach Manager at start up		
☐ Search required		
Optional comment		
OK	NetWare(R)...	Cancel Help

Figure 118. SNA Local Node Parameters on OS/2 Workstation

1. Enter the network ID and local node name for the OS/2 workstation with OS/2 Access feature.
2. Enter the local node ID if required by your network.
3. Enter the local node alias name if required by your applications.

8.3.3 Executing TN62TEST.EXE from the OS/2 Command Line

```
Before calling TN62:
opcode = AP_START_TN62
return code = (0x0000, 0x00000000)
fqcp_name = USIBMRA.SERVERB
cp_alias =
ip_address_mask = 0.0.0.0
vcbP->connection_retry_secs = 0
vcbP->unacked_dg_retry_secs = 0
vcbP->unsent_dg_retry_secs = 0
vcbP->inactivity_timer_secs = 0
vcbP->comwait_secs = 0
domain_name_suffix =

After calling TN62:
opcode = AP_START_TN62
return code = (0x0000, 0x00000000)
fqcp_name = USIBMRA.SERVERB
cp_alias = SERVERB
ip_address_mask = 0.0.0.0
vcbP->connection_retry_secs = 300
vcbP->unacked_dg_retry_secs = 10
vcbP->unsent_dg_retry_secs = 3
vcbP->inactivity_timer_secs = 30
vcbP->comwait_secs = 30
domain_name_suffix = sna.ibm.com
```

Figure 119. Result of the C:\tn62test.exe START USIBMRA.SERVERB Command

On the OS/2 command prompt, type: tn62test.exe START USIBMRA.SERVERB followed by Enter.

Note: This command is case sensitive, so the START VERB and FQCP-NAME must be in uppercase.

8.3.4 TCP62 Status Monitoring

Query Services	
Workstation Type	: Server
Default configuration	: TN62SRVB
Active configuration	: TN62
Service	Status
=====	
Kernel	ACTIVE
SNA Services	ACTIVE

Figure 120. Result of CMQUERY Command on Server-B

Note that we made the DUMMY configuration called: TN62SRVB on Server B. TCP62 then modified this configuration and made another configuration file called: TN62. This is now the active configuration used by eNetwork Communications Server for OS/2 Warp for TCP62 support.

	Link Name	Name	DLC #	Partner FQName	TG#	Type	HPR	Sess	State
*	@ANYAN	\$ANYNET	0	\$ANYNET.\$GWCP		LEN		0	Active

Figure 121. Result of CMLINKS Command on Server B before Executing the APING Command

As can be seen from the above figure that there are no sessions active between Server B to any other node before the APING command is executed.

8.3.5 APING from Server B to OS/2 Workstation

```

IBM APING version 2.43.3c  APPC echo test with timings.
Licensed Materials - Property of IBM
(C) Copyright 1994,1995 by IBM Corp. All rights reserved.

Allocate duration:                      625 ms
Program startup and Confirm duration:   1086 ms

Connected to a partner running on: OS/2

      Duration      Data Sent      Data Rate      Data Rate
      (msec)        (bytes)        (KB/s)         (Mb/s)
      -----
              8             200             24.4             0.195
              8             200             24.4             0.195
Totals:       16             400             24.4             0.195
Duration statistics:  Min = 8   Ave = 8   Max = 8

```

Figure 122. Result of APING from Server B to OS/2 Workstation

8.3.6 TCP62 Status Monitoring after Executing the APING Command

	Link Name	Name	DLC #	Partner FQName	TG#	Type	HPR	Sess	State
*	@ANYAN	\$ANYNET	0	\$ANYNET.\$GWCP		LEN		2	Active

Figure 123. Result of CMLINKS Command after the APING Command

As can be seen from the above figure, there are now two sessions active between Server B and the other node, after the APING command is executed.

8.3.7 TN62TEST.EXE Command Syntax

This is the syntax that must be followed if using the TN62TEST.exe program.

Usage:

TN62TEST [flags] operation

where flags is one or more of:

- q quiet mode - the only thing written to stdout is the blank-delimited primary and secondary return codes in hex. Useful for automated testing.

and operation is one of:

START fqcp_name
[cp_alias
[domain_name_suffix
[ip_address_mask
[connection_retry_secs
[unacked_dg_retry_secs
[unsent_dg_retry_secs
[inactivity_timer_secs
[conwait_secs]]]]]]]

STOP

DEFINE_LOCAL_LU lu_name
[ip_address_mask
[register_to_dns
[lu_alias]]]

DEFINE_PARTNER_LU partner_lu_name
[ip_address_mask
[partner_ip_address
[partner_hostname
[plu_alias
[plu_un_name
[max_mc_ll_send_size
[conv_security_ver
[parallel_sess_supp]]]]]]]
[alt_alias ...]]]]]]]

Notes:

- the parameters are positional. If the nth parameter is specified, then parameters 1 through n-1 must also be specified.
- omitted trailing parameters take their default values.
- the default value for a name is specified by '.'.
- the default value for a number is specified by 0.
- the default value for an IP address/mask specified by 0.0.0.0

Figure 124. The Syntax of the TN62TEST.EXE Command

See 8.2.13, "TCP62 Return Codes" on page 185 for information on primary and secondary return codes. A common return code is shown in Figure 125 on page 195.

The return code has the following syntax:			
Return Code: (Primary Return Code, Secondary Return Code)			
If you encounter the following return codes:			
Before Calling TN62:			
=====			
Return Code: (0x0000, 0x00000000)			
After Calling TN62:			
=====			
Return Code: (0x0100, 0xca010000)			
This means that you have supplied the wrong parameters to the TN62TEST.EXE program.			
Primary Return Code	Meaning	Secondary Return Code	Meaning
-----	-----	-----	-----
0x0000	Normal	0x00000000	Normal
0x0100	Parameter Error	0xca010000	Invalid CP Name

Figure 125. Some Useful Primary and Secondary Return Codes

8.3.8 RSP File Produced by TCP62

Note that not all of this file is shown due to its large size, however the most relevant information regarding TCP62 is highlighted and listed below:

```
* Installation keywords:
;
LOCAL_CP = (
    NAME = USIBMRA.SERVERB
    ;
    ;
    BRANCH_EXTENDER_SUPPORT = 0
    SEARCH_REQUIRED = 0
)
;
;
;
SNA_DEFAULTS = (
    ;
    DEFAULT_ROUTING_PREFERENCE = 1
    ;
;
;
ANYNET=(
    SNASUFFIX=sna.ibm.com
    CRITICAL_WS=YES
    TCPWAIT_MINS=15
    CONNWAIT_SECS=30
    CONN_RETRY_SECS=300
    UNACKED_DG_RETRY_SECS=10
    UNSENT_DG_RETRY_SECS=3
    INACTIVITY_TIMER_SECS=30
    NODE_TYPE=1
)
```

Part 6. Dependent LU Support

Chapter 9. SNA Gateway in Branch Extender Nodes

eNetwork Communications Server for OS/2 Warp provides a full-function Systems Network Architecture (SNA) gateway. The gateway allows multiple LAN-attached workstations to access System/370 or System/390 hosts through one or more physical connections to one or more hosts. This helps reduce the cost per workstation of host connections. In this chapter we describe a sample configuration of the SNA gateway in a branch extender node (BrNN).

9.1 SNA Gateway Support

The eNetwork Communications Server for OS/2 Warp gateway supports the SNA protocols LU 0, 1, 2, 3, and dependent LU 6.2 (APPC). With the SNA over TCP/IP function, downstream workstations can now communicate with the SNA gateway over an IP network. The gateway also supports LU 0, 1, 2, or 3 to an AS/400 host using SNA pass-through. The AS/400 host passes the data through to a System/390 host. A gateway also acts as a protocol converter between workstations attached to a LAN and a host WAN link.

An SNA gateway enables supported workstation applications to access remote supported applications on a subarea network without requiring a separate direct connection to each host in each workstation.

The LUs defined in the gateway can be dedicated to a particular workstation or pooled among multiple workstations. Pooling allows workstations to share common LUs, which increases the efficiency of the LUs and reduces the configuration and startup requirements at the host. You can also define multiple LU pools, where each pool is associated with a specific application. And you can define common pools that are associated with multiple hosts. When a link is defined through the gateway between a workstation and the host, the LU is activated when the session is established and returned to the pool for access by other workstations when the session is ended.

You can also configure an SNA gateway to automatically stop a session after a specified period of inactivity if there are other workstations waiting. This procedure helps increase the number of shared logical units that are available.

In addition, an SNA gateway can support the transmission of network management vector transports (NMVTs) between the gateway and the host. For example, commands coming from the NetView program in the host are received in the gateway and can be passed to and used by another application on the gateway, such as IBM LAN Network Manager.

Each host views the SNA gateway as an SNA PU 2.0 node, supporting one or more LUs per workstation. As far as the host is concerned, all LUs belong to the SNA gateway PU. The SNA gateway can have multiple host connections simultaneously and can direct different workstation sessions to specified hosts. However, only one host (and it must be on a link with a control point PU) can act as the focal point, and the control point name is appended to all NMVTs routed through the gateway.

To the supported workstations, the SNA gateway looks like an SNA PU 4 communications controller and forwards such host requests as BIND and

UNBIND. The workstation LUs are not aware of the SNA gateway. The SNA gateway, however, is aware of all LUs at the workstations.

9.2 SNA Gateway Configuration

In this chapter we provide a sample configuration of an SNA gateway in a branch extender node (BrNN) connected to an APPN network. In this scenario, a dependent LU requester (DLUR) PU is required in order to transport the 3270 emulation sessions over the APPN network to a VTAM APPN system.

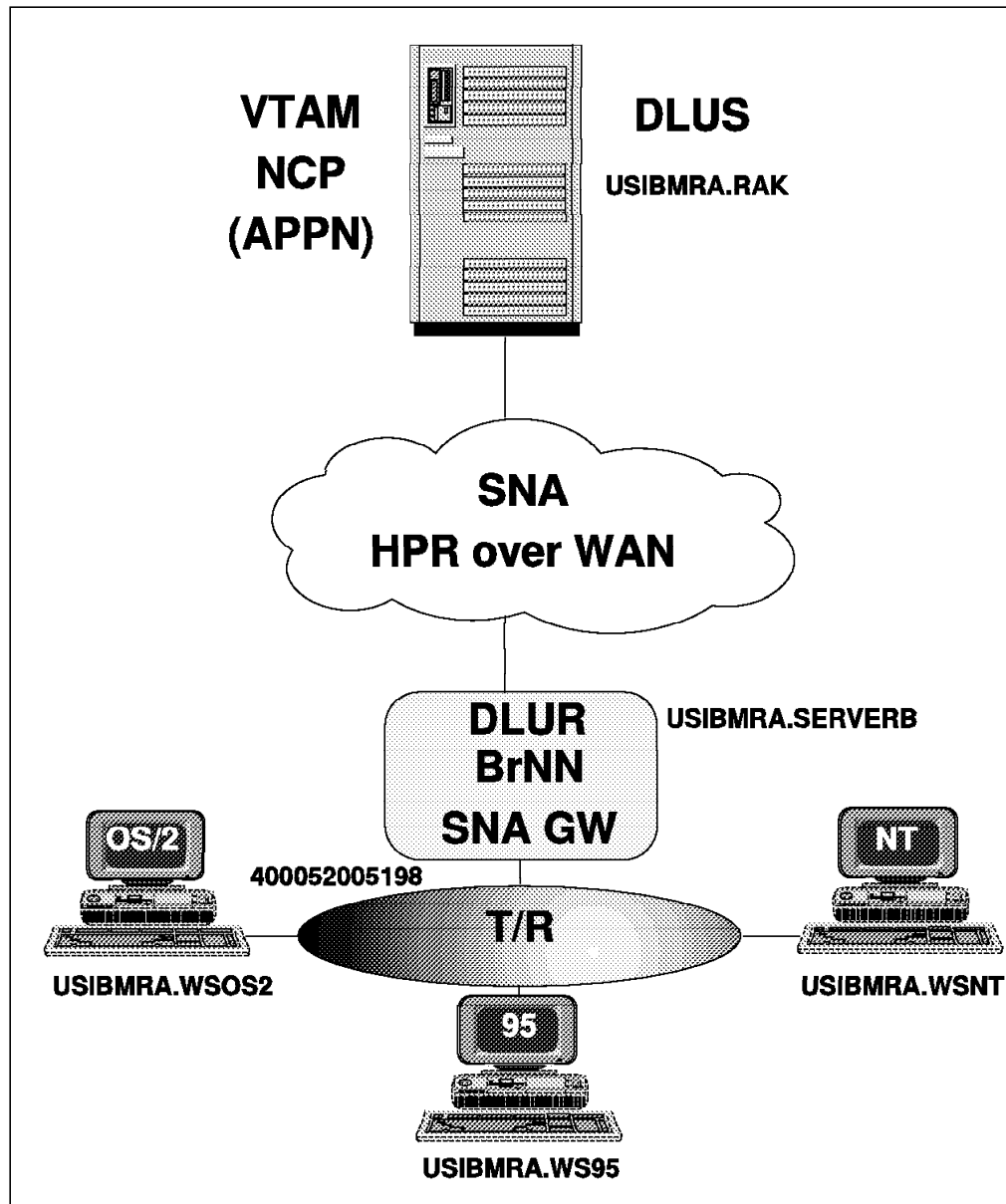


Figure 126. SNA Gateway Scenario

We configure the SNA gateway in a branch extender node (BrNN) for the scenario shown in Figure 126.

To configure the SNA gateway function in a branch extender node (BrNN), we recommend the following steps:

1. Define a local node and make sure you enable the node as a BrNN.
2. Enable BrNN on any uplinks or connections. For example, if the BrNN is connected to a VTAM network node, you should enable this uplink for BrNN.
3. Configure DLC adapters, which are used for the host and workstation links
4. Define host links, which are used to describe the SNA gateway connections to one or more hosts. A host link can be either a subarea connection or a DLUR PU when using an APPN network.
5. Define workstation LUs, which are the logical connections between the workstations and the gateway. Downstream workstations can be either explicit or implicit nodes.
6. Configure upstream dedicated LUs or host LU pools.

You access the gateway profiles by selecting **Configure...** from the Gateway pull-down menu in the eNetwork Communications Server for OS/2 Warp configuration window as shown in Figure 127.

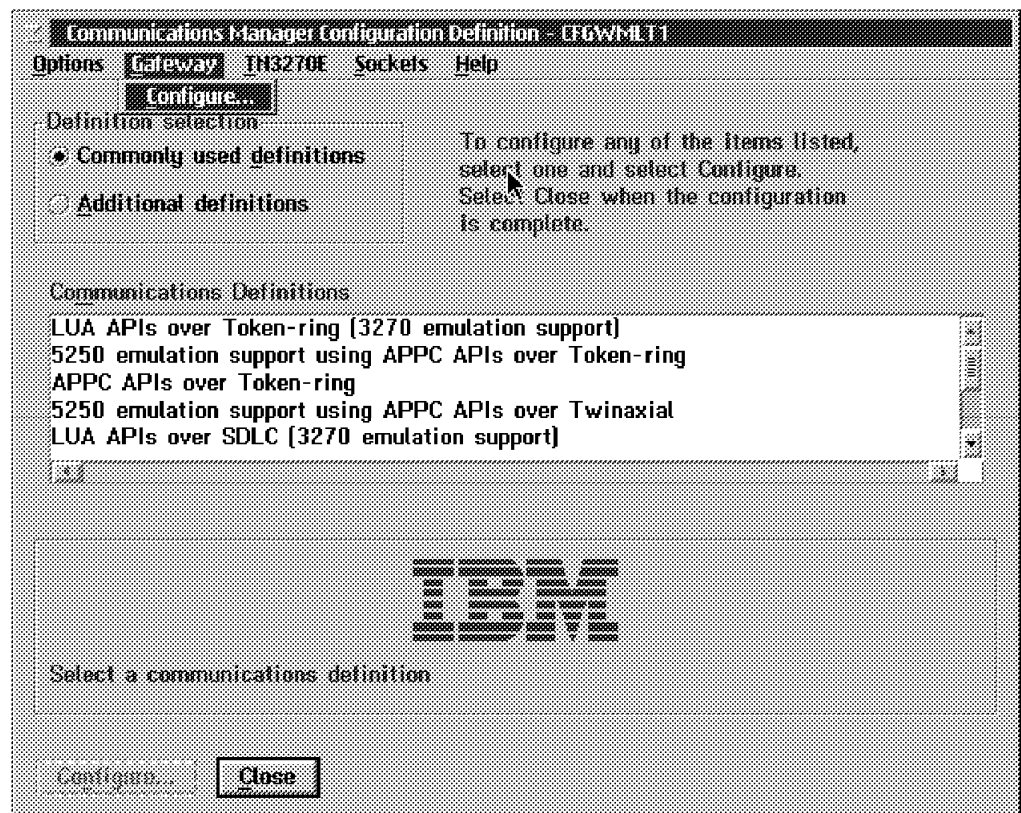


Figure 127. SNA Gateway Configuration

By selecting the gateway configuration, you obtain the list of profiles you need to configure in order to support the selected options. Note that some of the profiles are marked Required, as shown in Figure 128 on page 202.

The Communications Server SNA gateway lets an OS/2 workstation act as a communication controller or gateway between multiple workstations and one or more SNA hosts. Each workstation is connected to the gateway by a link, which can provide access to multiple logical units (LUs). Each subarea host link or a DLUR PU can support up to 254 dependent LUs.

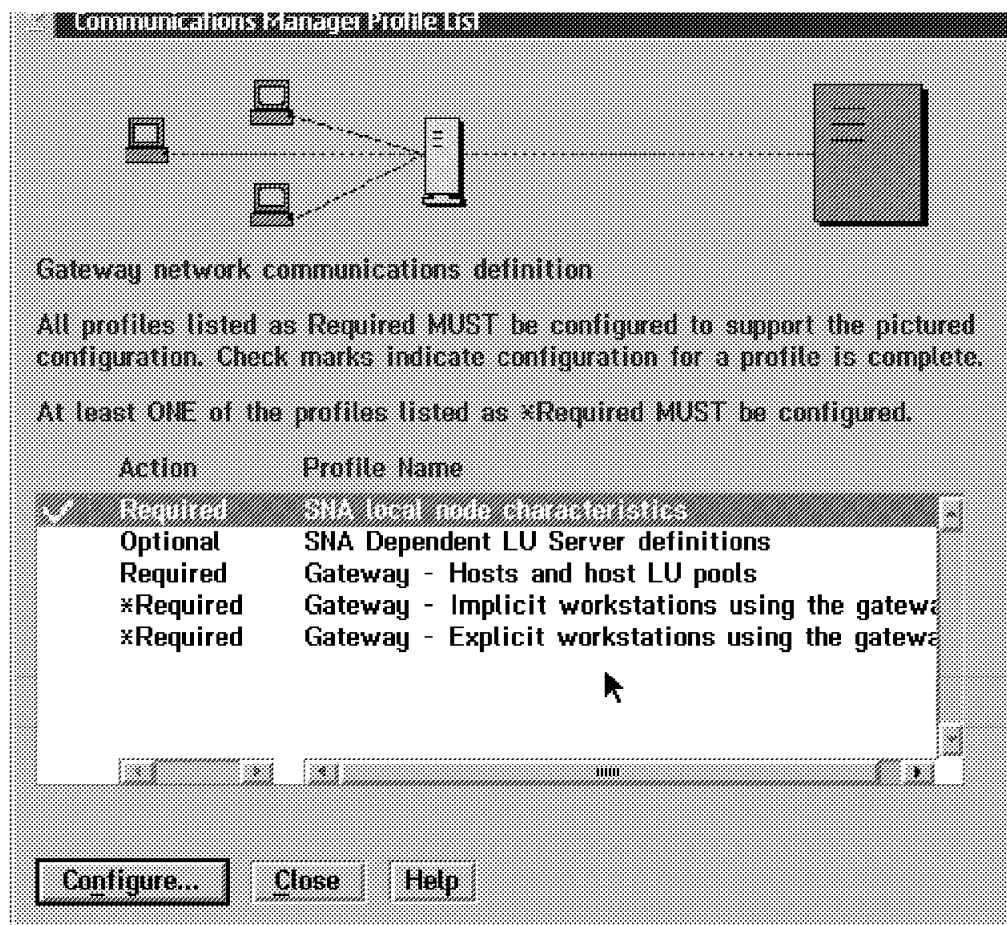


Figure 128. SNA Gateway Profiles

The profiles for the SNA gateway are:

- Gateway - Hosts and host LU pools
- Gateway - Implicit workstations using the gateway
- Gateway - Explicit workstations using the gateway

While configuring a profile, use the online help for a description of the parameters.

9.2.1 SNA Gateway - Hosts and Host LU Pools

This profile defines the hosts and host LU pools that the workstation can connect to. Before you can create the SNA gateway host definition, you must configure the SNA local node characteristics profile.

Obtain the following information from your network administrator:

- To define a gateway host link, obtain the connection information required for the DLC used to connect to the host.
- To define a host LU pool, obtain the start NAU address and the end NAU address for each pool to be created.

You can specify the same pool on several host definitions, creating a common pool across multiple hosts. To use the common pool, specify an asterisk (*) on the Define gateway workstation panel.

You can delete a host link and all associated host LU pools or you can delete individual pools that are part of the gateway host LU pool definition. If you delete a host LU pool, the pool no longer exists, and any workstations that were in the pool can no longer access the SNA gateway from the pool.

If workstation LUs are defined for a pool, delete those LUs that access the pool prior to deleting the pool.

9.2.2 SNA Gateway - Implicit Workstations Using the Gateway

This profile defines a generic workstation LU definition on a gateway for a workstation to have access to a host LU pool. Instead of explicitly creating a definition for every workstation using the gateway, you can use the implicit workstation definition.

Define at least one gateway host LU pool before defining implicit gateway workstations that use the gateway definition.

To create an implicit workstation definition from a gateway, obtain the workstation NAU address for the implicit definition from your network administrator.

You can specify an * for the host link name for the implicit workstation, indicating that the workstations can use any host link that has the pool name defined.

9.2.3 SNA Gateway - Explicit Workstations Using the Gateway

This profile defines a workstation LU definition at the gateway for a workstation to have access to one or more hosts. This LU can be pooled or dedicated.

To create or change an explicit workstation definition from a gateway, obtain the following information from your network administrator:

- The connection information required for the DLC used to connect to the workstation
- The workstation NAU addresses
- The host NAU address, if the LU is dedicated

Following are some points to remember when configuring the SNA gateway function:

- If the profile that you select does not have a prerequisite profile, the selected profile definition window appears. If the profile that you selected requires another profile to be configured first, a message appears that directs you to the appropriate profile.
- You can change several workstation definitions at once using response files and the CID process. For information on changing response files, refer to the online *Network Administration and Subsystem Management Guide* located in the Information Notebook.
- Automatic logoff is helpful in an LU pool. For example, if 8 LUs must service 15 LU sessions on 15 workstations and if all 8 LUs are active, the other 7 workstation LUs cannot log on to the host.

- By having a workstation automatically log off after a user has not used an LU for a certain period of time, other workstation users have a chance to log on to the host. A workstation is logged off only when all sessions are in use and the gateway receives a session request from another workstation.
- If you select All configured DLCs from the DLC type list in the Implicit Workstations Using the Gateway window, a workstation can connect to the host through any configured DLC on the gateway. If you select a specific DLC type, you are limiting access to only those workstations using that DLC type.
- You can have multiple links to the same host or links to more than one host. You must define host link names for multiple connections to one host or connections to more than one host link.
- The starting point and ending point of the range cannot fall within or overlap the NAU address ranges defined for any other pool. You can define more than one range for a pool.
- The Maximum number of link stations parameter in the LAN DLC profile limits the number of workstations that can be supported by the gateway. The default value for this parameter might not be sufficient for your gateway. In most cases, you need to increase it.
- The total number of link stations you use for LAN Gateway and the number of link stations you use for Communications Server cannot exceed the number of link stations configured in MPTS.

9.2.4 BrNN Local Node Configuration

You always need to configure the local node characteristics of your server. The network ID defines the name of your Advanced Peer-to-Peer Networking (APPN) network.

Note

A branch extender node (BrNN) is not required to have the same network ID as its network node server (NNS).

Together, the network ID and the node name (fully-qualified control point name) uniquely identify a node within an interconnected network environment. The network ID and the node name also uniquely identify a node for receiving error logs and network management alerts.

Local Node Characteristics

Network ID: USIBMRA

Local node name: SERVERB

Node type:

- ☐ End node
- ☒ Network node
- ☒ Branch extender support

Local node ID: [hex] 05D 00002

Local node alias name: serverb

Maximum compression level: NONE

Maximum compression tokens: 0 (0 - 30400)

☒ Activate Attach Manager at start up

☐ Search required

Optional comment:

OK NetWare(R)... Cancel Help

Figure 129. Local Node Configuration of a BrNN

9.2.5 DLUR PU - Configuration

If your server is connected to an APPN network, you need to configure one or more DLUR PUs to be used by the SNA gateway.

Note

If your server is connected to a VTAM subarea system, it needs to be adjacent to the VTAM system and the connection (host link) defines the required PU. In this case the PU is referred to as a subarea PU.

Dependent LU Server Parameters

Link name: DLURPU01

Fully-qualified dependent LU server name: USIBMRA . RAK

Local PU name: SRVBPU

Node ID (hex): 05D DDD11

Optional fully-qualified backup dependent LU server name: USIBMRA

☒ Maximum activation attempts: [1 254]

☒ Activate at startup

Optional comment: DLUR PU to VTAM

OK Apply Cancel Help

Figure 130. Dependent LU Server Parameters - DLUR PU Configuration

In our scenario, the BrNN is connected to an APPN network and therefore we define a DLUR PU as follows:

- Link name identifies the DLUR PU you are defining.
- Fully-qualified dependent LU server name is the primary DLUS name in your network.

Note

Check with your VTAM coordinator for the proper DLUS name in your network. It is normally the fully-qualified VTAM CP name.

- Local PU name. It is used internally for event logging only and therefore any name can be entered.
- Node ID. This is the unique DLUR PU node ID (IDNUM/IDBLK).

Note

Check with your VTAM coordinator for the proper node ID you can use for each DLUR PU you define.

- Backup fully-qualified dependent LU server name. Leave blank if you do not have a backup DLUS.
- Maximum activation attempts. Select this option if you want to limit the number of activation attempts.
- Activate at startup. You will normally select this option to initiate the DLUR/DLUS pipe from the server at startup time.

9.2.6 Host Links

A host link can be either a connection to a VTAM subarea, a connection to another SNA gateway (cascaded gateways) or a DLUR PU if the server is connected to an APPN network.

In our scenario, the server is connected to an APPN network and therefore the host link is the previously defined DLUR PU we configured in Figure 130 on page 206.

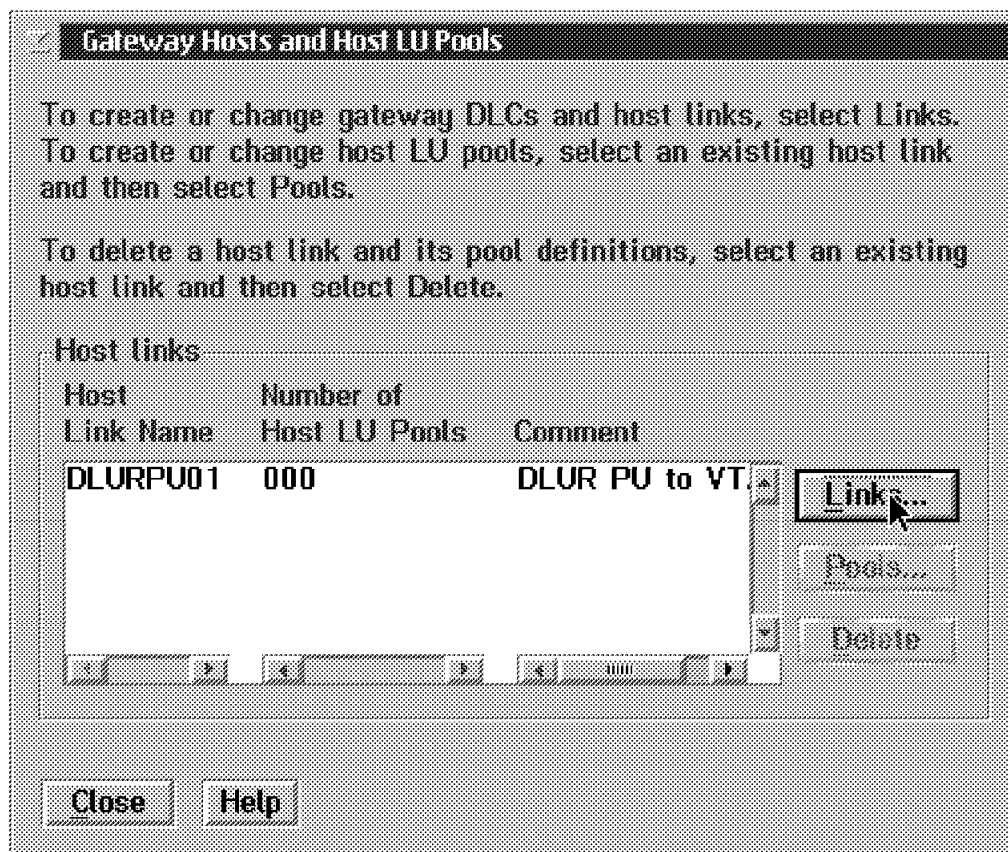


Figure 131. SNA Gateway Host Links and DLUR PUs

9.2.7 Host LU Pools

You must define a host LU pool to support downstream implicit workstations. For explicit workstations, host LU pools are optional and you can use dedicated host LUs.

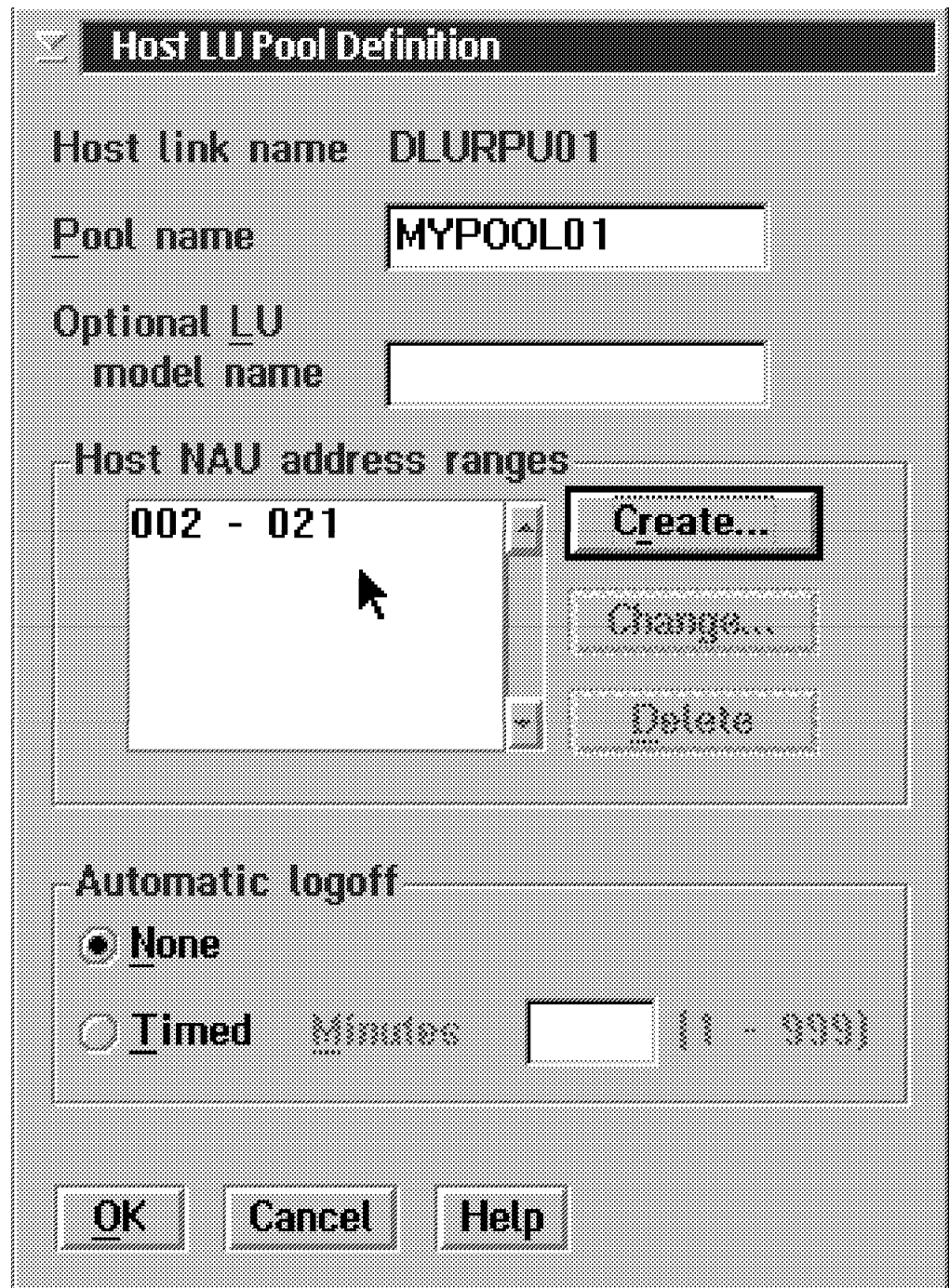
In our scenario we select **Pools...** to define a host LU pool to be used by the downstream implicit workstations.

When you create a host link name, you provide information based on the type of connection to the host.

When you create a host LU pool, you provide the following information:

- Host link name. In our scenario it is the DLUR PU name.
- Pool name. Provide a name to identify the host LU pool.

- Host network addressable unit (NAU) address ranges. In our scenario we have 20 LUs (range 2 - 21) associated with the DLUR PU.
- Whether the pool includes auto-logout, and the amount of inactive time until automatic logout.



Host LU Pool Definition

Host link name DLURPU01

Pool name

Optional LU model name

Host NAU address ranges

002 - 021	Create...
	Change...
	Delete

Automatic logout

☒ None

☐ Timed Minutes (1 - 999)

OK Cancel Help

Figure 132. SNA Gateway - Host LU Pool Configuration

Select **Create...** to define a logical unit (LU) pool for a host link. A window appears in which you can type a pool name and the LU addresses at the host for the pool. This indicates to the gateway which LUs at the host the workstations can use to communicate with the host.

You can create one or more LU pools to accommodate groups of workstation users who require access to the host. If a group of workstation users needs host access for the first time, you may want to create an LU pool for them, rather than allowing more workstations access to an existing LU pool.

To create gateway LU pools for a host link, select **Create....**

Another window appears, prompting you for more information.

9.2.8 Downstream Workstations

There are two types of gateway-supported workstations:

- Implicit workstations
- Explicit workstations

Implicit workstations are much easier to configure than explicit workstations, but they can use only pooled LUs. Instead of defining a link to each workstation using the gateway, you define a host pool (or pools) and configure the DLCs for the connections that the workstations are using. You then configure a model LU definition for each workstation network addressable unit (NAU) that connects to the gateway.

For example, if each workstation to an Ethernet LAN has two sessions that support 3270 emulation configured with NAU addresses 2 and 3, then you would configure two implicit LU definitions in the gateway: one for NAU 2, another for NAU 3. In this example, each time a workstation connecting to the gateway over Ethernet is logged on, a link is dynamically created, and the two LUs for NAU 2 and 3 are allocated from the host pool. If an asterisk (*) is specified for the host link name when the workstation is configured, this link can be to any host that has that pool defined. An asterisk is the default for the host link name for implicit workstations.

For implicit workstations, users connecting to the gateway need to know only the adapter address of the gateway data link control (DLC) that is configured for the implicit workstations and what NAU values have been defined on the gateway.

Explicit workstations have defined destination addresses over a particular DLC type (for example, token-ring network and SDLC). To configure explicit workstations, you must know the destination address of each workstation and also define a logical link to the gateway for each workstation.

LUs that are defined for explicit workstations can be pooled or dedicated. Pooled LUs are configured at the SNA gateway and grouped in pool classes. Pooled LUs are not dedicated to a specific workstation. Dedicated LUs are dedicated to a specific workstation, configured at that workstation, and known to the host by an LU local address at the host.

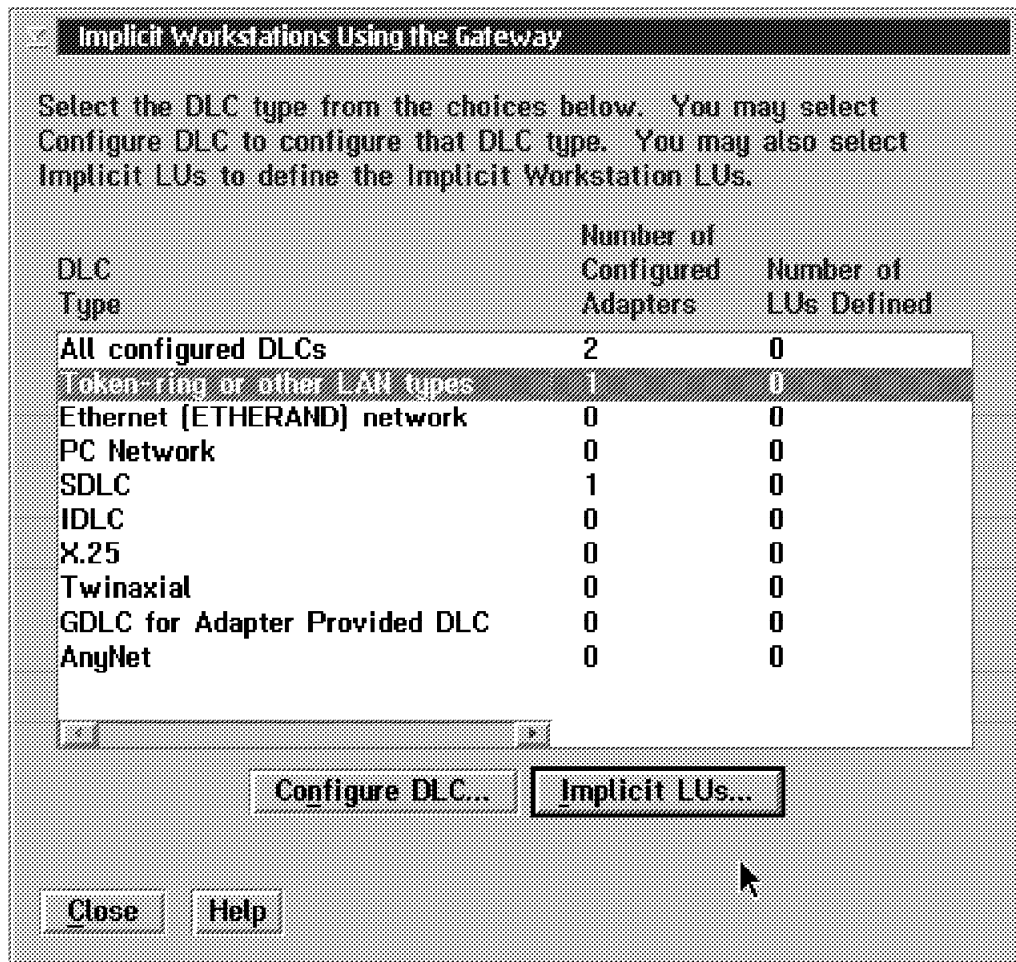


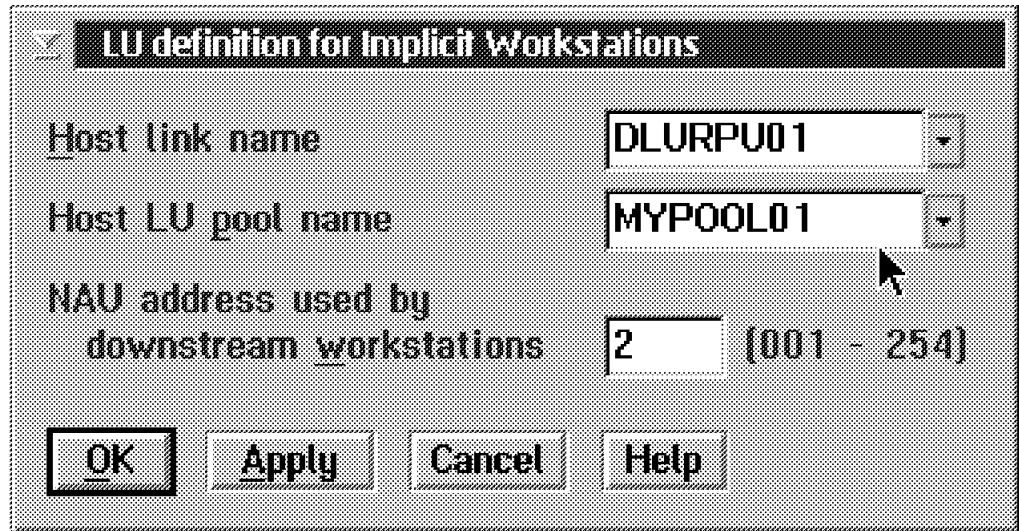
Figure 133. Implicit Workstation Configuration

The DLC Type List displays the possible DLC types, which can be used to configure your gateway connections. Select the DLC type you want to use or select All configured DLCs if you want to make your workstation LUs independent of DLC type.

The Number of Configured Adapters field states the number of configured adapters configured for this DLC type. For all configured DLCs this field states the total number of configured adapters.

Note: If the Number of Configured Adapters field states "0" you can use Configure DLC... to configure the selected DLC type.

The Number of LUs Defined field states the number of LUs defined for this DLC type.



The image shows a dialog box titled "LU definition for Implicit Workstations". It contains three input fields: "Host link name" with the value "DLURPU01", "Host LU pool name" with the value "MYPPOOL01", and "NAU address used by downstream workstations" with the value "2" and a range "(001 - 254)". At the bottom, there are four buttons: "OK", "Apply", "Cancel", and "Help".

Host link name	DLURPU01
Host LU pool name	MYPPOOL01
NAU address used by downstream workstations	2 (001 - 254)

Buttons: OK, Apply, Cancel, Help

Figure 134. Implicit Workstations - Downstream LUs

The Host link name field displays the various host links defined at the gateway. You can define a workstation logical unit (LU) for any of these host links. If an asterisk (*) appears in this field, the LU uses any available session with the specified host LU pool name. This window is used only for pooled workstation LUs and it is not used for dedicated LU connections to the host.

Chapter 10. TN3270E Server in Branch Extender Nodes

In this chapter, we show you how to configure the TN3270E server in a branch extender node and how the server provides VTAM access to the downstream telnet 3270 workstations.

10.1 TN3270E Scenario

As an example, we provide a working configuration for the scenario shown in Figure 135.

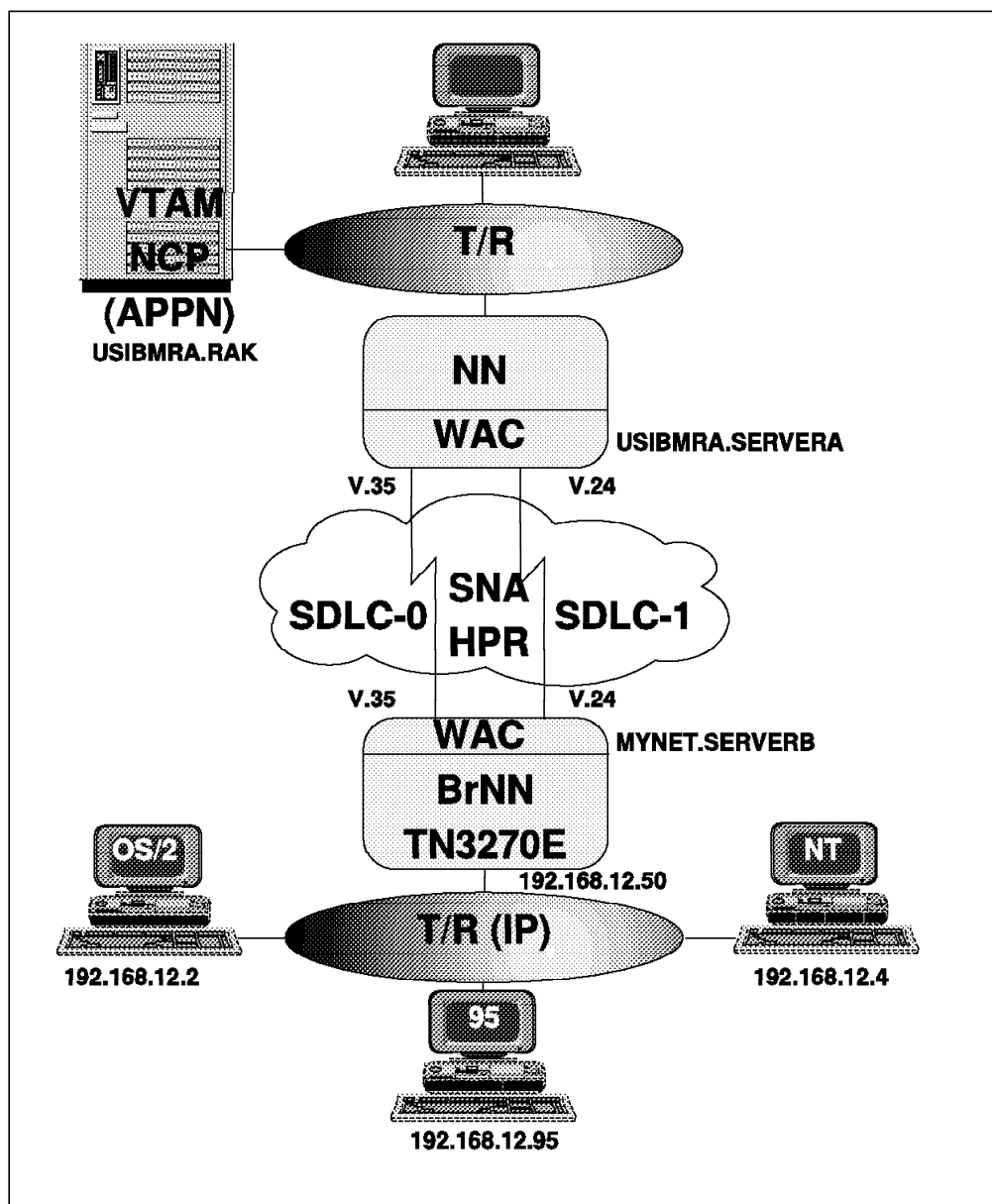


Figure 135. TN3270E Server in a BrNN - Scenario

10.2 TN3270E Server Configuration

To configure the TN3270E server, enter CMSETUP from the OS/2 command line or click the configuration setup icon in the eNetwork Communications Server for OS/2 Warp folder. You will then enter the name of your configuration for this scenario to proceed with the TN3270E server configuration.

In the main configuration panel, you will now choose the TN3270E server parameter profile by selecting **Configure** from the TN3270E pull-down menu as shown in Figure 136.

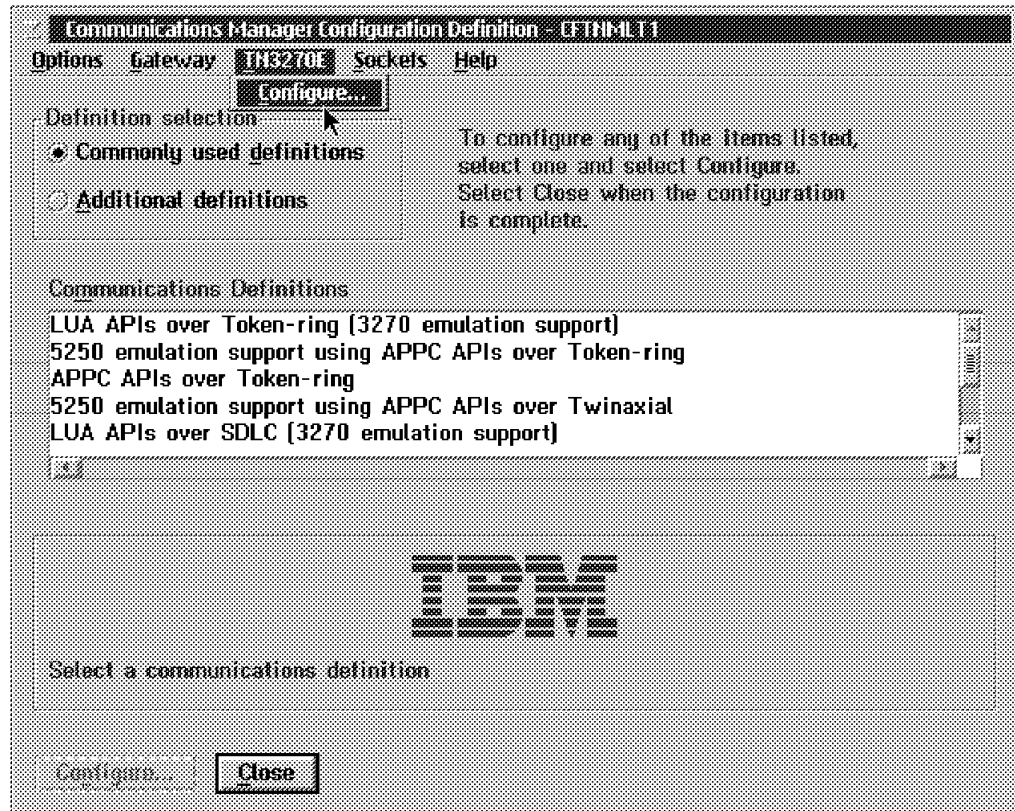


Figure 136. TN3270E Server Configuration

For the TN3270E server common scenario, you will be required to configure the local node and the TN3270E parameters (see Figure 137 on page 215).

Upstream to the VTAM system the connection is via SNA using the LU application (LU-A) interface. For this reason, the connection can be either a subarea connection by defining a host link or it can be an APPN connection using a dependent LU requester (DLUR) PU.

In our scenario, our VTAM system is enabled for APPN and provides a dependent LU server (DLUS) to support the dependent LU sessions. Therefore, we configure a DLUR PU by using the optional profile SNA Dependent LU Server definitions.

Note

If required you will configure multiple DLUR PUs in order to support a large number of dependent LUs. In SNA a DLUR PU supports up to 254 LUs and each dependent LU supports only one session.

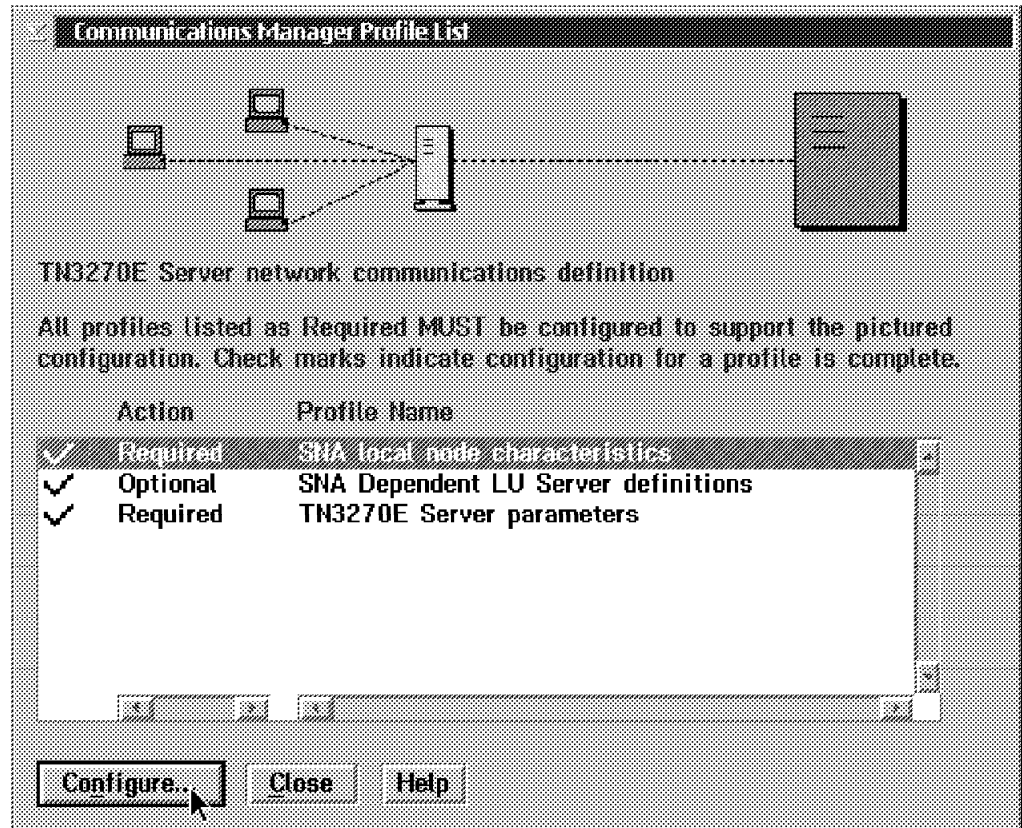


Figure 137. TN3270E Server Configuration Options

10.2.1 SNA Local Node Characteristics

Start with SNA local node characteristics profile.

- Enter the network ID and local node name.
- Select network node with branch extender support (BrNN).

Note

The TN3270E does not require the node to be a BrNN. Actually the node can be an EN, a NN, or a BrNN. It can also have a LEN connection to a VTAM subarea network, to an SNA gateway or to another node using an AnyNet SNA over TCP/IP link.

- The local node ID and the local node alias name are optional in our scenario.
- If you need data compression, enter the required definitions.

In our scenario, the node is configured as shown in Figure 138 on page 216.

Local Node Characteristics

Network ID: MYNET

Local node name: TNSRVB

Node type:

- ☐ End node
- ☒ Network node
- ☒ Branch extender support

Local node ID: [hex] 05D 00002

Local node alias name: tnsrvb

Maximum compression level: NONE

Maximum compression tokens: 0 (0 - 30400)

☒ Activate Attach Manager at start up

☐ Search required

Optional comment:

OK NetWare[R]... Cancel Help

Figure 138. Local Node Characteristics

10.2.2 SNA Dependent LU Server Definitions

We now configure a single DLUR PU to support the upstream SNA sessions. We are not including the adapter or connection to a network node. Therefore, to configure links to network nodes see DLC configuration in Chapter 5, "APPN and High-Performance Routing" on page 87.

The following options are configured to have a DLUR PU (see Figure 139 on page 217):

- Link name. Enter a name to identify this DLUR PU.
- Fully-qualified dependent LU server name. This is the network ID and the CP name of the primary DLUS in your VTAM system.
- Local PU name. This name is used for event logging.
- Node ID. Obtain a valid node ID you can use from your VTAM administrator.
- Fully-qualified backup dependent LU server name. Enter the name of a backup DLUS if any. Otherwise, leave it blank.
- Maximum activation attempts. Use this option only if you want to limit the maximum number of activation attempts.
- Activate at startup. You will normally check this box to allow your node to start the DLUR/DLUS connection from the server at start up time.

Dependent LU Server Parameters

Link name: DLURPUTN

Fully-qualified dependent LU server name: USIBMRA . RAK

Local PU name: PUTN3270

Node ID (hex): 05D B0012

Optional fully-qualified backup dependent LU server name: USIBMRA . RAS

☐ Maximum activation attempts [1 - 254]

☒ Activate at startup

Optional comment:

OK Cancel Help

Figure 139. Dependent LU Server Parameters

10.2.3 TN3270E Server Parameters

We now configure the TN3270E server parameters. Note that the TN3270E server will use the previously configured DLUR PU and therefore you do not need to create a link in this case as shown in Figure 140 on page 218.

Note

Use the create link option in TN3270E server configuration if your node is connected to a VTAM subarea system to create a TOHOST link.

You can either configure explicit or implicit telnet 3270 workstations. In our scenario, we decided to configure implicit workstations only. This means that a workstation can use any LU from the DLUR PU to establish a session.

In our scenario we configure the TN3270E server as follows:

- Host link name. Select the DLUR PU name configured in Figure 139.
- Create link. Not required for a DLUR/DLUS connection. We assume that the link to a network node has been already configured.
- Number of implicit workstation definitions. We will support up to 6 workstations.

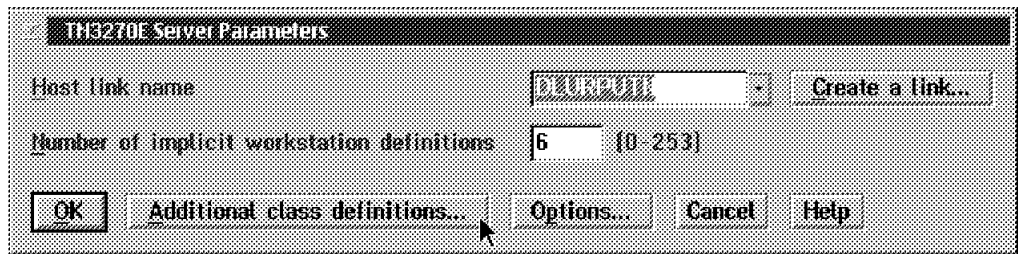


Figure 140. TN3270E Server Parameters

Select **Additional class definitions..** to create LU-As for this scenario as shown in Figure 141. If using printers, you will associate printers in this panel.

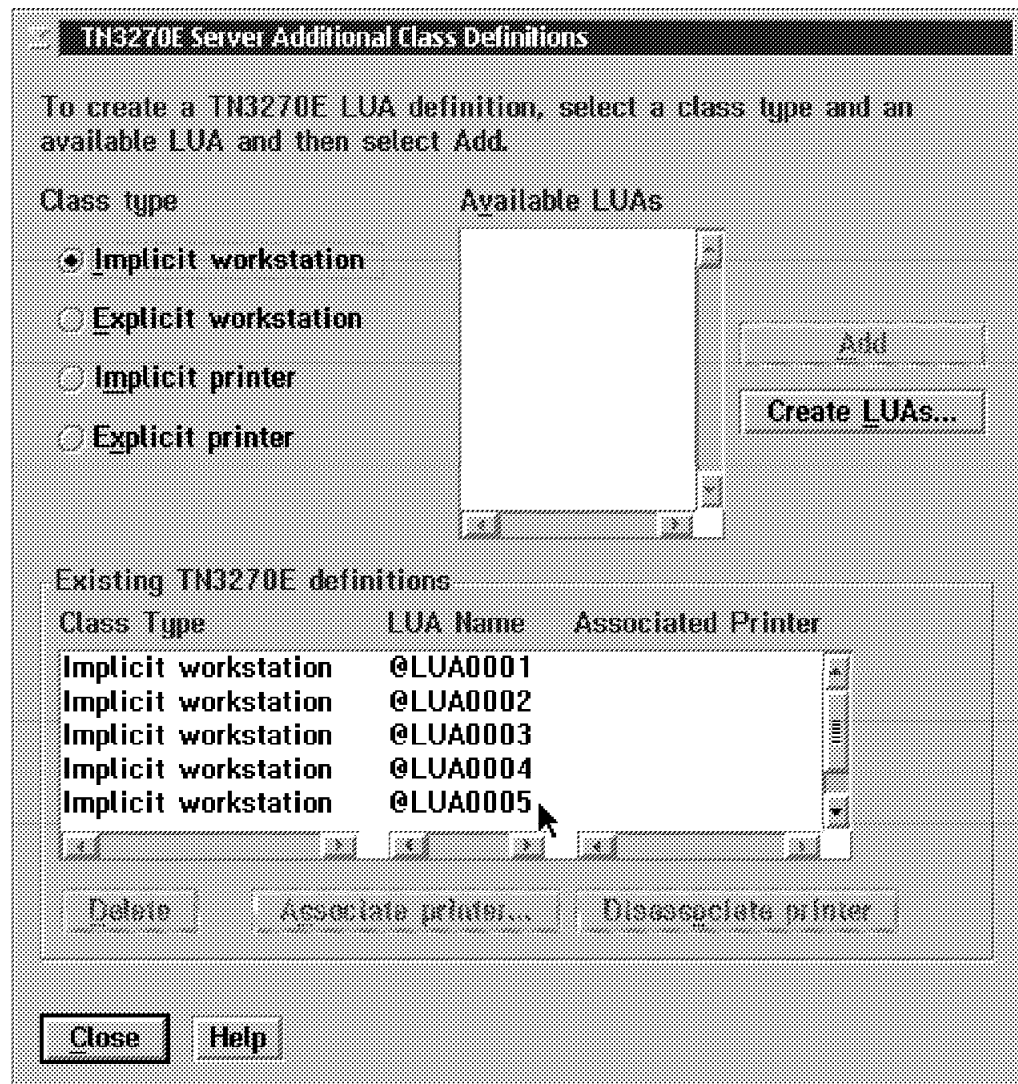


Figure 141. TN3270E Server Additional Class Definitions

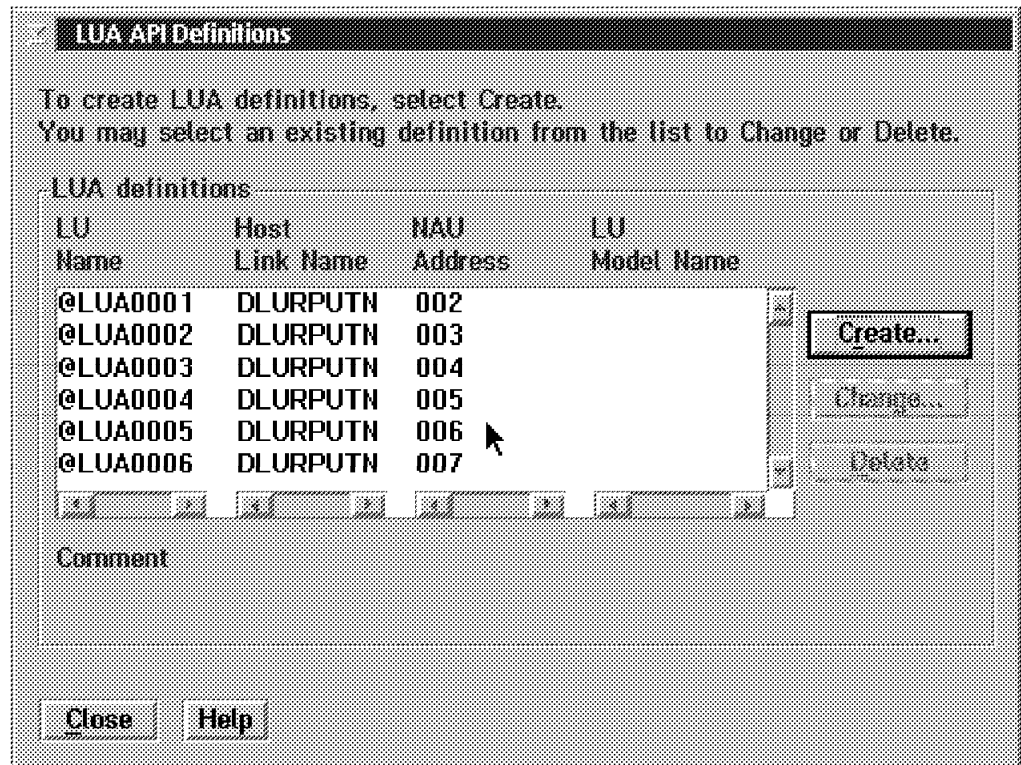


Figure 142. LUA API Definitions

In the TN3270E server parameters panel, you select **Options** to configure any optional parameters if required (see Figure 143).

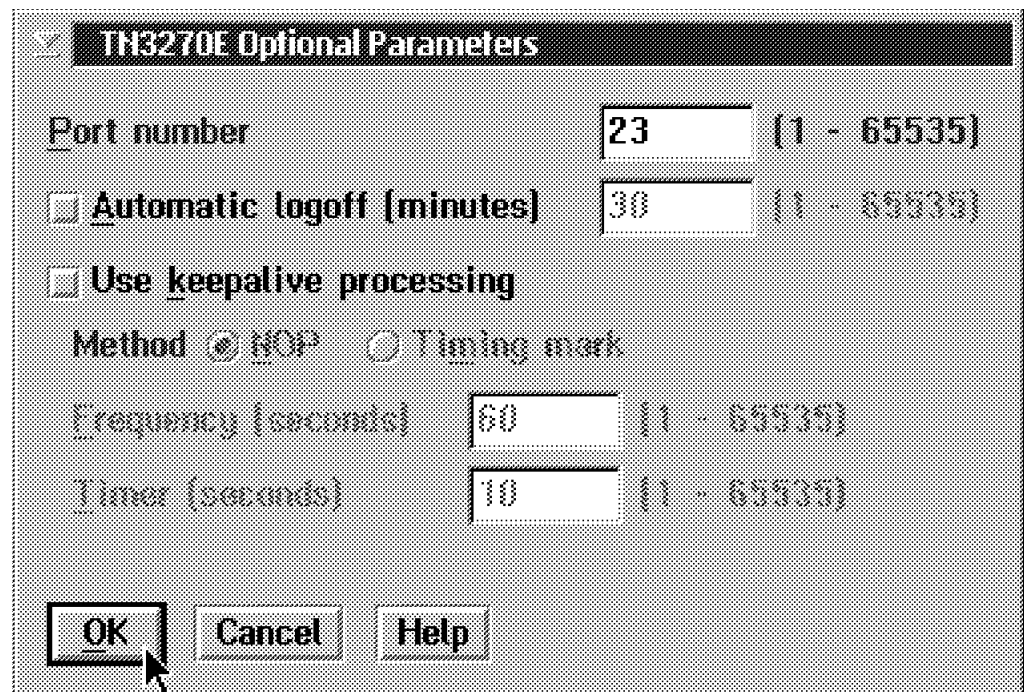


Figure 143. TN3270E Optional Parameters

- The Port number is the number of the port that the TN3270E application will use.

Normally, the TN3270E Server uses port 23, which is the default.

Note

If you are running telnetd in this machine, you must select another port for TN3270E Server. To use another port, type a number between 1 and 65535. You should avoid port numbers that you know are used by other applications, because if two applications use the same port number, then one of the two applications will fail.

- The Automatic logoff is the amount of idle time to allow before the session is dropped.
- Select Use Keepalive processing if you want to automatically detect that a TCP/IP connection has been broken. You can choose to use either NOP or Timing mark. The command for either method is only sent when no traffic has been sent or received on a connection for the number of seconds specified in Frequency.

Select NOP to use the Telnet NOP command. Use NOP if you do not want to generate additional traffic on your TCP/IP network and you do not need connections freed immediately.

When you select NOP, if no traffic is sent or received for the number of seconds specified in Frequency, the server transmits data on the connection, which enables TCP/IP to detect that the connection has been broken. This could take several hours.

Select Timing mark to use the Telnet timing mark command. Use this selection if you need to free connections immediately and additional traffic on your network is acceptable.

When you select Timing mark, if no traffic is sent or received for the number of seconds specified in Frequency, the server requests that the client respond immediately. If the client does not respond within the number of seconds specified for Timer, the connection is closed.

10.3 Workstation Configuration

Client workstations are configured using the telnet 3270 access function. See Chapter 3, "Windows 95/NT Client Workstations" on page 33 and Chapter 4, "OS/2 Client Workstations" on page 59 for details on how client workstations are configured.

10.4 TN3270E Server Administration

There are several options you can use to monitor telnet 3270 activity in your server. In the eNetwork Communications Server for OS/2 Warp select the folder **Administration**.

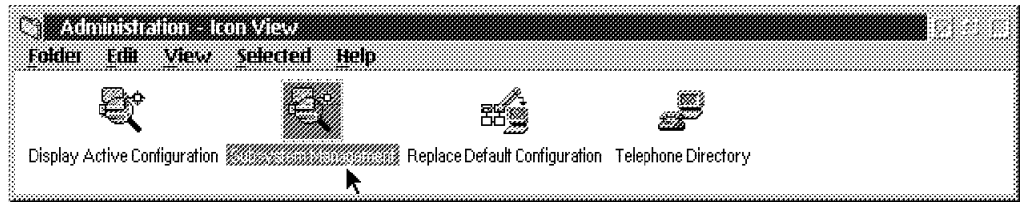


Figure 144. Administration Folder

In the Administration panel (see Figure 144), select **Subsystem Management** to verify that the TN3270E server has been started as shown in Figure 145.

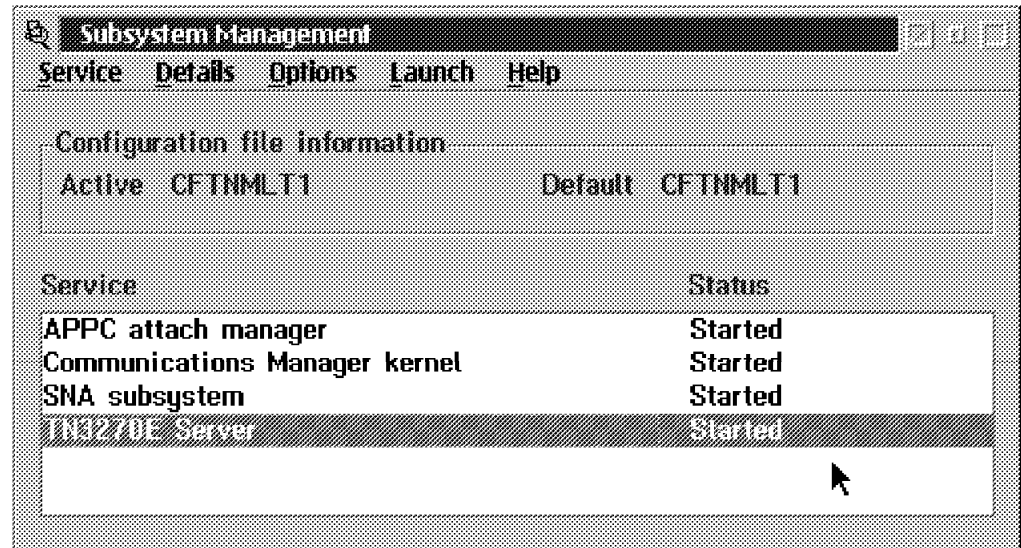


Figure 145. Subsystem Management - Main Panel

In Subsystem Management, select **TN3270E Server sessions** from the Details pull-down menu.

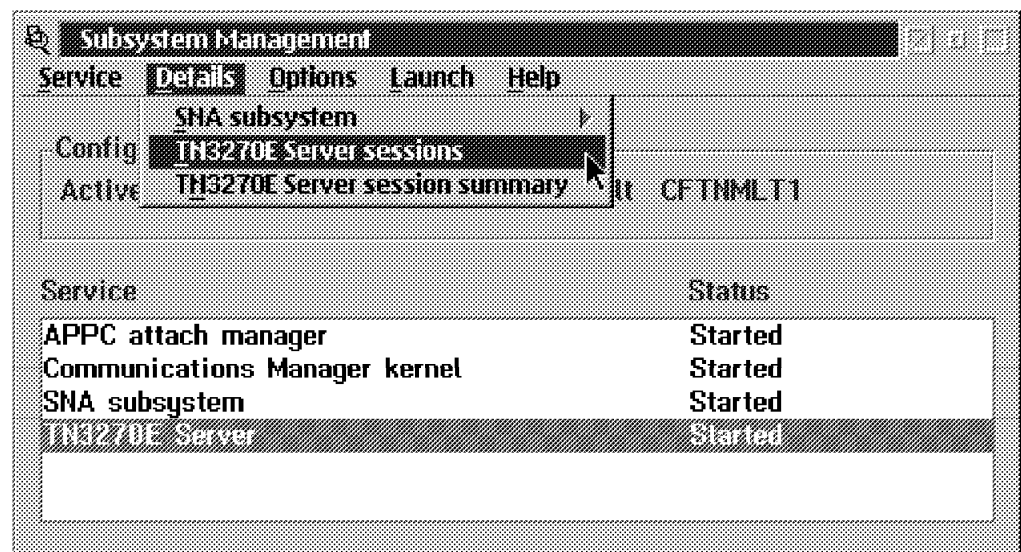
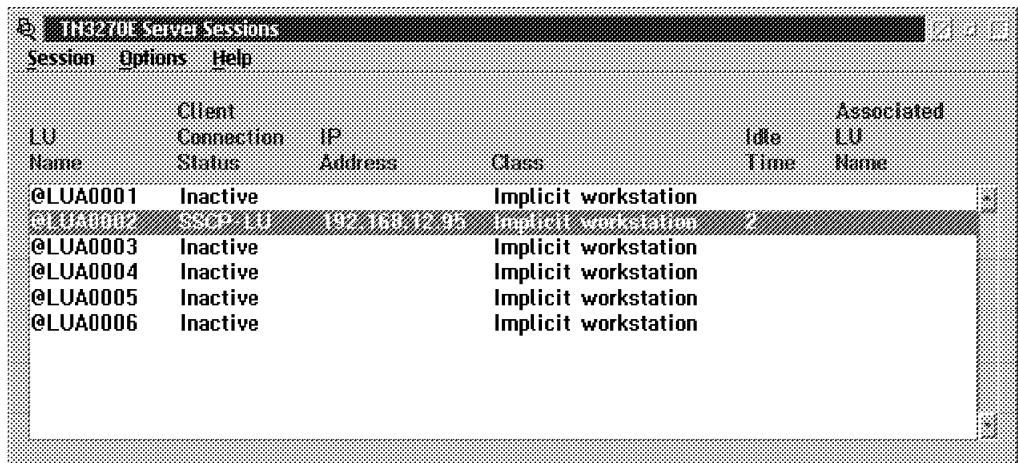


Figure 146. Subsystem Management - Details

The TN3270E Server Sessions window shows information about the sessions. For example:

- Connection Status
- Workstation IP Address
- Idle Time

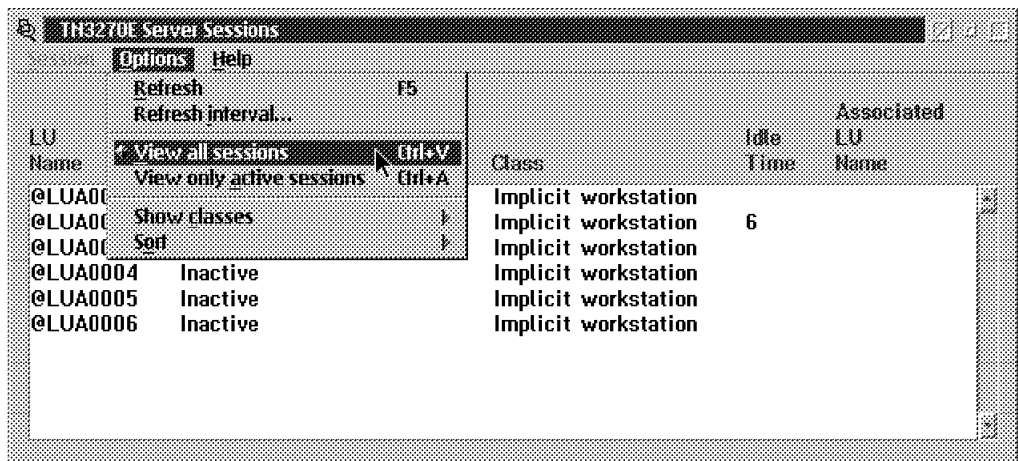


The screenshot shows the 'TN3270E Server Sessions' window with a menu bar (Session, Options, Help) and a table of sessions. The table has columns for LU Name, Client Connection Status, IP Address, Class, Idle Time, and Associated LU Name. The data is as follows:

LU Name	Client Connection Status	IP Address	Class	Idle Time	Associated LU Name
@LUA0001	Inactive		Implicit workstation		
@LUA0002	SSCP LU	192.168.12.95	Implicit workstation	2	
@LUA0003	Inactive		Implicit workstation		
@LUA0004	Inactive		Implicit workstation		
@LUA0005	Inactive		Implicit workstation		
@LUA0006	Inactive		Implicit workstation		

Figure 147. TN3270E Server Sessions

The TN3270E Server Sessions panel can be customized to view all sessions or only the active sessions by selecting these options from the **Options** pull-down menu (see Figure 148).



The screenshot shows the 'TN3270E Server Sessions' window with the 'Options' menu open. The menu options are: Refresh (F5), Refresh interval..., View all sessions (Alt+V), View only active sessions (Alt+A), Show classes, and Sort. The table of sessions is partially visible behind the menu:

LU Name	Client Connection Status	IP Address	Class	Idle Time	Associated LU Name
@LUA0001	Inactive		Implicit workstation		
@LUA0002	Inactive		Implicit workstation	6	
@LUA0003	Inactive		Implicit workstation		
@LUA0004	Inactive		Implicit workstation		
@LUA0005	Inactive		Implicit workstation		
@LUA0006	Inactive		Implicit workstation		

Figure 148. TN3270E Server Sessions - Options

From the Details pull-down menu in Subsystem Management you can also display a TN3270E server session summary.

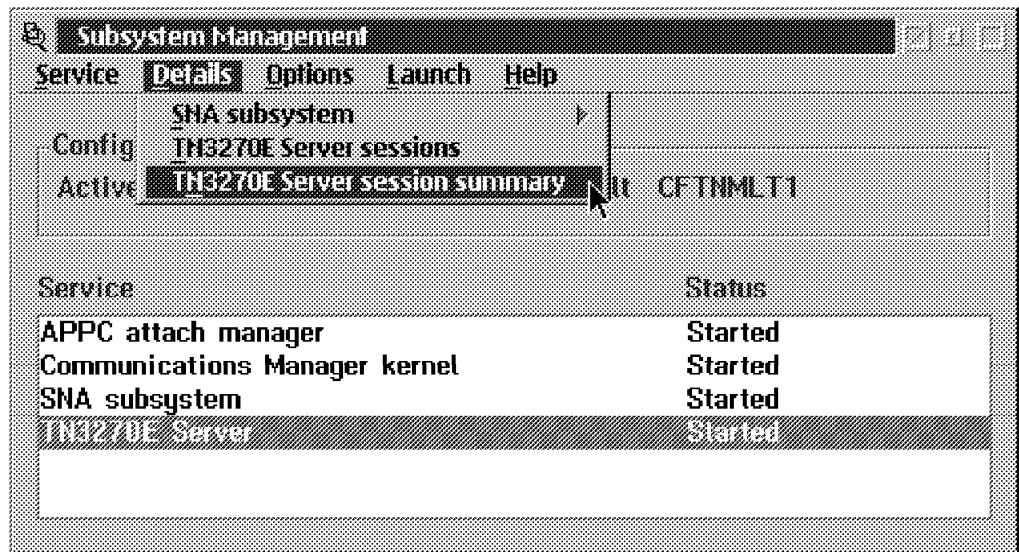


Figure 149. TN3270E Server - Session Summary

This option displays an information summary about the TN3270E server session. For a sample display see Figure 150.

The screenshot shows a window titled "TN3270E Server Session Summary Information" with a menu bar containing "Options" and "Help". The main content area displays a table with three columns: "Class Type", "Number of LUs Configured", and "Number of LUs In Use".

Class Type	Number of LUs Configured	Number of LUs In Use
Explicit workstation	0	0
Implicit workstation	6	1
Explicit printer	0	0
Implicit printer	0	0
Associated printer	0	0
-----	-----	-----
Total	6	1

Figure 150. TN3270E Server Session Summary Information

Chapter 11. Host On-Demand

IBM Host On-Demand is a Java-based Intranet/Web to SNA solution. It provides a high-performance, low-cost solution for Intranet and Web users who need occasional access to their central computer applications or databases from any Java-enabled end-user platform. Valuable centralized host information is now available to Web-centric end users. In this chapter we present an overview of the HOD installation process.

Note

The Host On-Demand product provided in eNetwork Communications Server for OS/2 Warp Version 5.0 is an entry level version. For more elaborate HOD support you should refer to the latest version of the HOD product.

11.1 Host On-Demand - Overview

Access is as simple as pointing and clicking on an SNA application hot link from within the user's Java-enabled Web browser. No customer programming, additional hardware, or previously installed client emulator is required.

Host On-Demand boasts a number of powerful features and benefits, such as:

- Emulator functions on demand through the dynamic download of the Host On-Demand Java applet
- Automatic session connection to your central computer
- End user customization of default session options
- Dynamic session hotlinks
- Two 3270 host sessions
- Resizable session windows with dynamically adjusted fonts
- Dual built-in 3270 function keypads
- Menubar with pull-down menu items
- Iconic toolbar
- Operator (OIA) display area
- End user and administrator help facilities

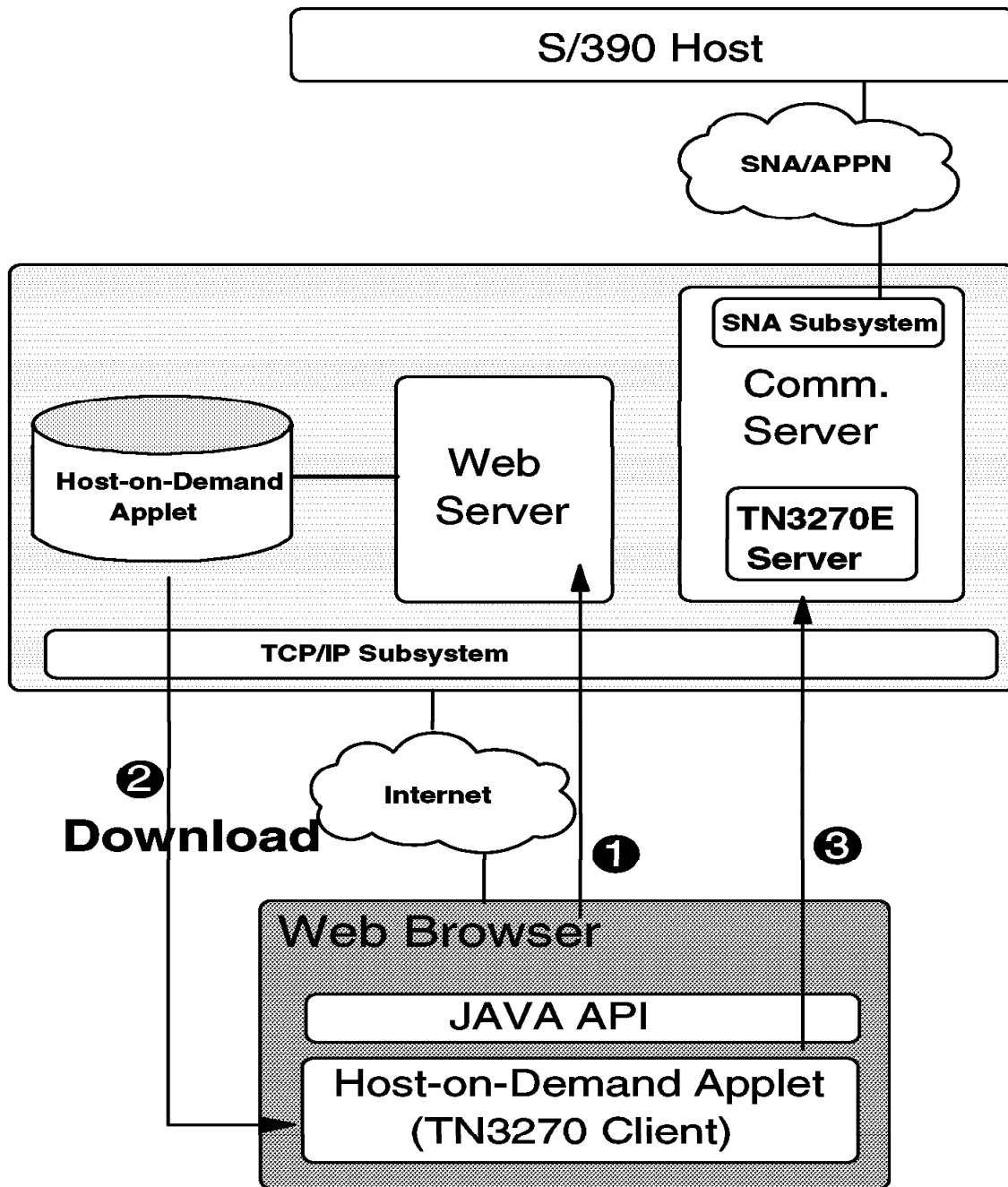


Figure 151. Host On-Demand Overview

Figure 151 illustrates the flow between a client and HOD server as follows:

1. The user connects to a Web server. Dependent on how this Internet server is configured, he or she either gets the Host On-Demand application by default or requests it specifically by entering a specific URL such as <http://servername/alias/he3270en.htm>, for example.
2. The server then downloads a Java applet to the browser which is a small standard Telnet 3270 application; this Telnet 3270 client will therefore run

with any industry standard Telnet 3270 server, in this case Communications Server for OS/2.

3. This application then contacts the same server and connects into the TN3270E server of a Communications Server residing on the same server machine as the Internet server. Note that it is mandatory at this point in time, that the TN3270E server reside on the same server as the Internet server; however, that may change in the future.

11.1.1 Required Software - Server

Host On-Demand is installed on a server and requires a Web server. IBM Communications Server and Host On-Demand do not include a Web server function. Any Web server capable of serving Java applets can be used in support of Host On-Demand.

Host On-Demand testing has focussed on Web servers from these companies:

- IBM
- Lotus
- Microsoft
- Netscape
- Novell

In our environment, we installed IBM Internet Connection Security Server (ICSS 4.2.1 or higher).

See Chapter 12, "Web-Based Administration" on page 239 for the Web page addresses for ICS information.

Note

Host On-Demand requires an IBM Communications Server to be installed in the same logical server as the Web server. This Version of Host On-Demand is for eNetwork Communications Server for OS/2 Warp Version 4.1 or later.

11.1.2 Required Software - Clients

Host On-Demand supports any Java-enabled platform with an industry standard Java virtual machine level of at least 1.02.

Our testing has determined that the following Web browsers interact best with IBM Host On-Demand:

- Microsoft Windows 95/NT
 - Netscape Navigator 3.0 (or later)
 - Microsoft Internet Explorer 3.0 (or later)
 - Notes Client 4.5
- IBM AIX 4.1.4
 - Netscape Navigator 3.0 (or later)
- IBM OS/2 4.0 with latest JDK
 - Netscape Navigator for OS/2 2.02

11.2 Host On-Demand Installation

To install Host On-Demand, run **Install** from the CD and the main screen will display.

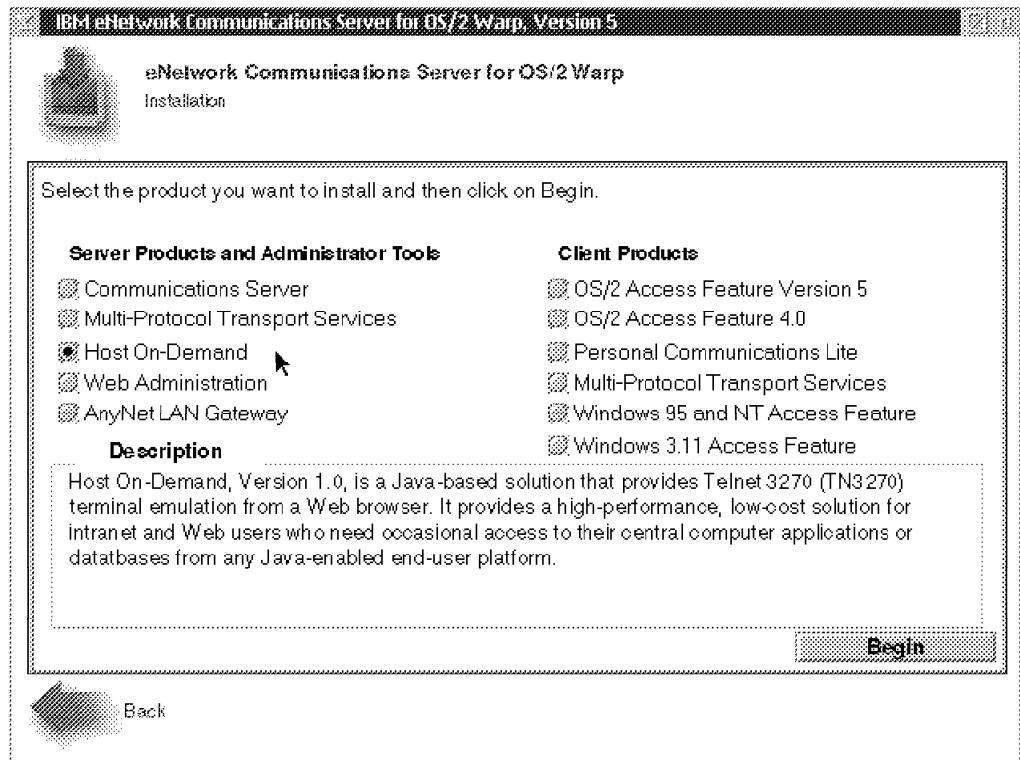


Figure 152. eNetwork Communications Server Installation

In the Server Products and Administrator Tools choose **Host On-Demand** and click **Begin**.

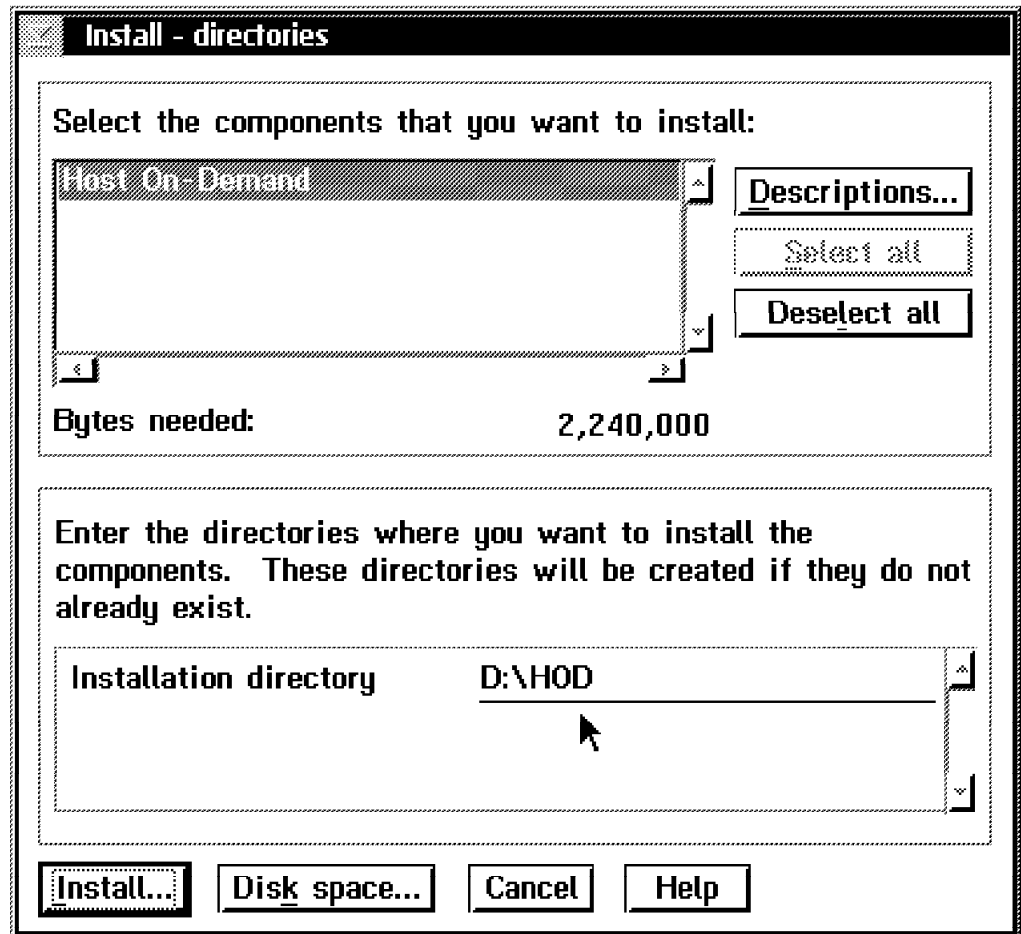


Figure 153. Host On-Demand - Installation Directory

On the Install-directories window, select the **Host On-Demand** component and click **Install**. Host On-Demand will be installed in the HOD directory.

11.2.1 ICSS Request Routing for HOD

Since HOD has been installed in the d:\HOD directory, you will need to create a pass in ICSS. This is accomplished by accessing the server using a browser (for example, Netscape). You will log on to the server and select the Configuration and Administration options as shown in Figure 157 on page 235.

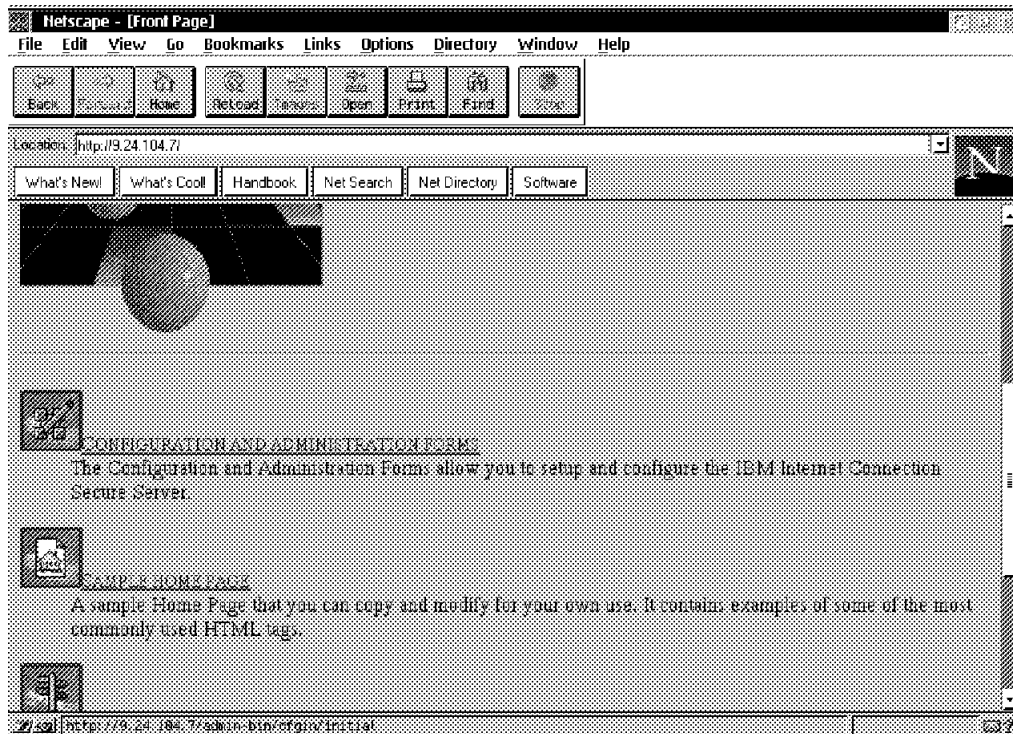


Figure 154. ICSS 4.2.1 - Configuration and Administration

You will select **Request Routing** and add the pass entry hod to point to the proper path where HOD was installed, as shown in Figure 155.

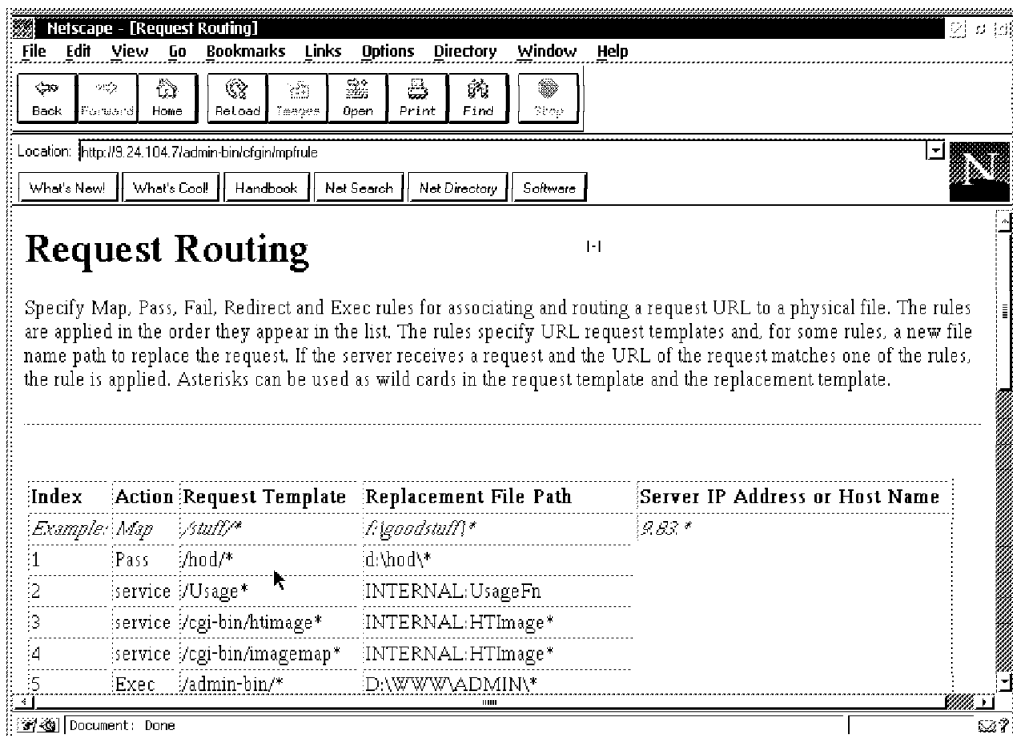


Figure 155. ICSS 4.2.1 - Request Routing for HOD

11.2.2 IBM Communications Server for OS/2 Configuration

To use the Host On-Demand product, IBM Communications Server must be installed and configured for TN3270 support. Sample TN3270 server configuration response files are available in the CD\RSPFILES\SERVER\TN3270S and CD\RSPFILES\SERVER\TN3270A directories. TN3270S.RSP is for a non-APPN server configuration. TN3270A.RSP is for an APPN/HPR configuration.

To configure IBM Communications Server to use a sample response file, follow the steps below:

1. Copy TN3270x.RSP to the CMLIB directory on the server.
2. To create a valid configuration file, type from an OS/2 command prompt:
`CMSETUP /R TN3270x.RSP`
3. To customize the SNA parameters for your network, run Communications Manager Setup from the Communications Server folder or from an OS/2 command prompt type: **CMSETUP /C TN3270x.RSP**.
4. Start IBM Communications Server with this configuration.

11.2.3 Host On-Demand Configuration and Additional Information

The readme.htm located in the directory where Host On-Demand is installed can be accessed from a Web browser. Refer to this readme for additional information on administration and configuration for Host On-Demand.

With a Web server and IBM Communications Server both installed, configured, and active, system administrators or Webmasters can establish a hyper-link to he3270en.htm and immediately begin using Host On-Demand through this link.

However, administrators may elect to customize the appearance and invocation mechanisms of IBM Host On-Demand by modifying the <param> tags in he3270en.htm. A variety of appearances may be presented to end users by customizing the he3270en.htm page. The following values can be customized.

- Title and masthead used on IBM Host On-Demand's invoking Web page
- Automatic connect to a fixed telnet server
- Automatic start as a separate window or as part of the browser

Refer to the comments in he3270en.htm and the sample*.htm sample files for additional details concerning invocation customization.

Administrators may also elect to customize the IBM Host On-Demand administrator pages headmnen.htm and headm1en.htm contained in the \en directory. These pages may be customized to provide pointers for your end users to your company or organization help desk or technical support personnel.

If you anticipate Microsoft Internet Explorer 3.0 users accessing IBM Host On-Demand, you may want to consider signing the Host On-Demand cabinet file (he3270ap.cab) with your company or organization's credentials. The Host On-Demand cabinet file has space reserved for your signature. Please refer to the Microsoft Web site for details concerning signing the Host On-Demand cabinet file.

11.2.3.1 Netscape Navigator 2.0

Set "Image" and "Audio" to "no" on the startup panel when running Netscape 2.0. Otherwise, it may cause severe performance and display problems. If you want to run with these options, please consider upgrading to Netscape 3.0.

Session windows running under Netscape 2.0 will not accept a "dot" (that is, decimal point or period) keystroke input directly from the keyboard. If this becomes a problem, please consider upgrading to Netscape 3.0.

There are JVM problems in painting the Host On-Demand session window when running Netscape 2.0 under Windows/NT 3.51. If the following conditions become a problem, please consider upgrading to Netscape 3.0:

- Graphical toolbar entries are not repainted.
- Graphical toolbar entries overlap with 3270 presentation space.
- The cursor does not paint on new host screens.

11.2.3.2 Microsoft Internet Explorer

The resizable session window function does not always work for MSIE 3.0 on the Win95 and NT platforms. If you are using Microsoft Plus!, in your Display Properties, ensure that the Show window contents while dragging visual setting is disabled.

The dotted IP address notation cannot be used to specify your TN3270 server under earlier builds of MSIE 3.0. This is a known problem of MSIE 3.0.

11.2.3.3 IBM OS/2 Client Environment

Users of the OS/2 appletviewer and Netscape for OS/2 should periodically check <http://www.ibm.com/java/> for information on, and updates to, your environment.

11.2.3.4 Performance

Host On-Demand runs in the Java Virtual Machine (JVM) environment of your browser or operating system, which provides its own garbage collection function for memory. Memory consumption for an active session in normal usage is about 1 MB RAM. When the session is closed, about 1/8 of the memory is released immediately by Java and the rest freed gradually over a period of time. For normal usage, this should not be a problem. However if you continuously start up and shut down sessions over an extended period of time, it could cause memory resource shortages and appear to hang your system.

You may establish sessions to your host for an extended period of time. When you resume activity on a long-idle host session, you may see a slight hesitation. This is normal because the host may have timed out and the desktop resources may have been consumed by the other activities. Host On-Demand will regroup and get your host session back to normal operational state quickly.

The debugging facility is initially activated by setting the applet's Debug parameter="yes" in he3270en.htm. A Debug submenu item will then appear in the Help menu item and a Debug icon will appear on the toolbar of the session window. Only use this facility under the direction of your system administrator or IBM support personnel; certain debug settings significantly impact session performance.

11.2.3.5 Miscellaneous

End user help information is available in two forms: browser HTML files and *What is it* support. If you are executing IBM Host On-Demand outside a browser (for example, using an appletviewer), the HTML-based help facility is not directly available to you from Host On-Demand session windows. *What is it* support, selected from either the menubar or toolbar, is always available; to use, select *What is it*, then click the item for which you want help.

The initial Host On-Demand htm page provides hot session links to sessions running in separate windows. Clicking on an active link will bring you to that active session window. However, if the session window is minimized, no action will occur. This behavior is a restriction for many Java virtual machines.

The F10 key on the keyboard is often reserved for Java Virtual Machine use. When you need to use F10 to interact with host applications, select the PF10 key on the keypad. Further, many Java virtual machines do not recognize all keyboard key values (for example, the "Insert" key). As a result, no corresponding direct keyboard support for those keys can be provided. Select these keys from the keypad. Refer to the Keyboard help information for the list of keyboard keys supported.

A session window can be separate or embedded in the invoking browser window. If it is a separate window, it can be resized by pointing the mouse, clicking at the border and dragging. If the window is embedded, it is not resizable and the scroll bar of the browser may be needed to view the entire session window.

To minimize the amount of scrolling when running with option separate window No, have your system administrator adjust the applet size in the Host On-Demand invoking Web page to your display screen resolution size, for example, 800x600 or 640x480. Turn off the options in your browser that show the browser toolbar, status area, location, directory buttons, etc. This will give you more 3270 display space and minimize the need for scrolling.

This release of Host On-Demand does not implement character attributes. This means that character-by-character control of colors, reverse video, etc, is not implemented. Colors, reverse video, etc, is available at the field level.

Base platforms may cause window repainting problems sometimes, such as when the toolbar icons or keypad becomes clipped when scrolling an embedded window up and down in a browser. This problem can be corrected by forcing a repainting.

11.2.4 Host On-Demand Session

After the Host On-Demand installation you have to start the eNetwork Communications Server for OS/2 Warp configuration with TN3270E Server.

For more information about TN3270E Server configuration, see Chapter 10, "TN3270E Server in Branch Extender Nodes" on page 213.

From the workstation try to connect by Web browser entering:

<http://serverb/hod/he3270en.htm>

Where serverb is the server hostname and hod is the directory where the Host On-Demand files are located.

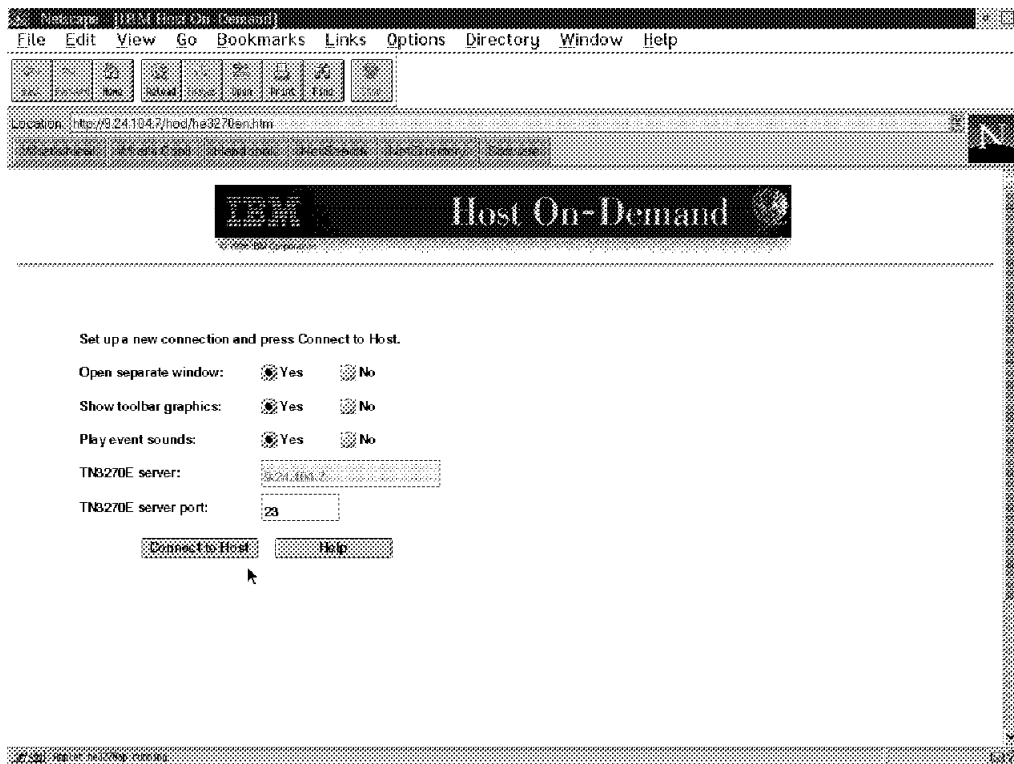


Figure 156. Host On-Demand Home Page

Once the user selects **Connect to Host**, this triggers the applet to contact the TN3270E server; if the TN3270E server is available and host LUs are available, you should receive the 3270 application welcome message as shown in Figure 157 on page 235.

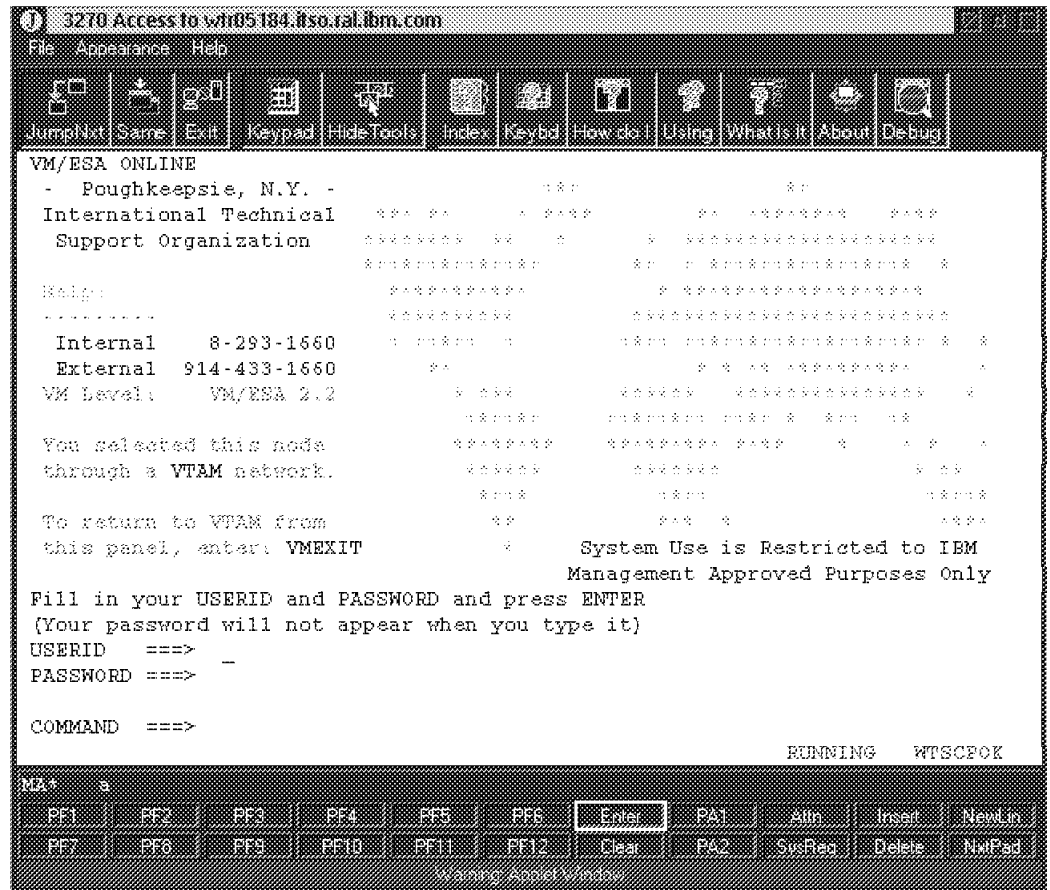


Figure 157. Host On-Demand Session

If the browser is not Java-enabled, this is recognized by the Web server and a message is issued to the browser indicating that a Java-enabled browser is required to use Host On-Demand.

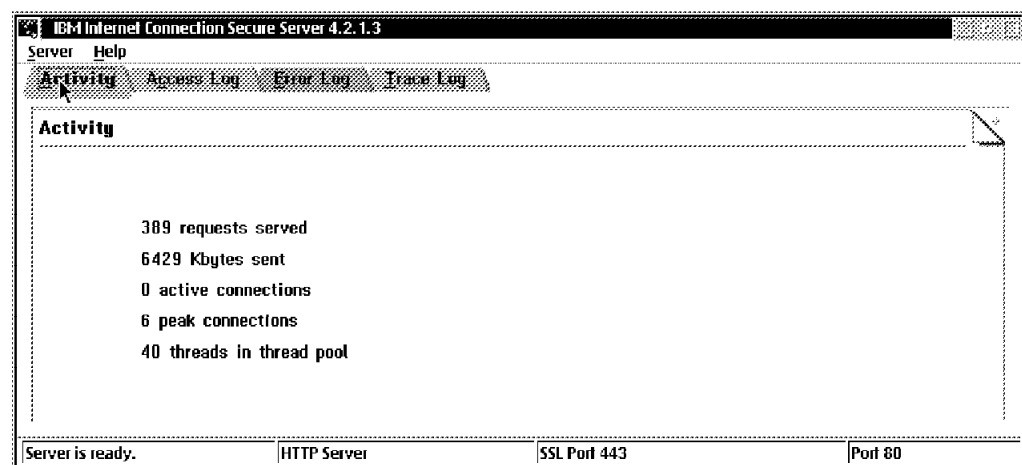


Figure 158. ICSS 4.2.1 - Activity

In case you cannot obtain a HOD session, we suggest you look at the trace log provided by the IBM Internet Connection Secure Server. A sample trace log display is shown in Figure 159 on page 236.

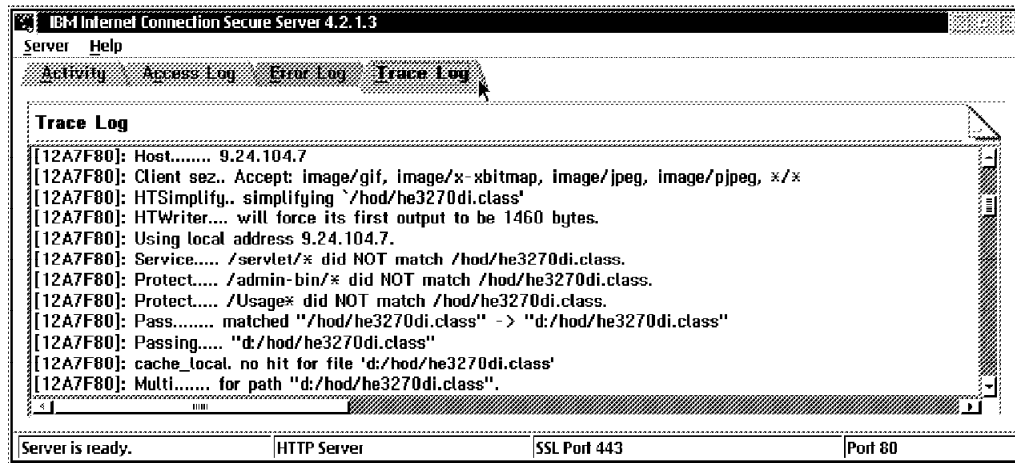


Figure 159. ICSS 4.2.1 - Trace Log

Part 7. Operations and Administration

Chapter 12. Web-Based Administration

The Web Administration feature provides the Web administration function for eNetwork Communications Server for OS/2 Warp. You can use this feature to manage Communications Server over an intranet or the Internet. Using a Web browser, an administrator can query node status, obtain information about resources, modify resources, display configuration files, display message logs and perform other administrative tasks. In this chapter we describe what Web software is required, how to install Web Administrator and how to get started using Web Administrator for OS/2 from a Web browser client.

12.1 Required Software

Web Administration requires a Web server running with Communications Server for OS/2 Warp and a Web browser running on any system on your intranet or the Internet. Communication between the Web browser and Web server over your network requires TCP/IP. If your network does not support TCP/IP, you can use of the Sockets over SNA function provided with Communications Server, with reduced function. Other required software includes:

On the Server

- IBM Communications Server for OS/2 Warp, Version 4.0 or higher.
 - IBM Internet Connection Security Server (ICSS 4.2.1 or higher).

A free copy of IBM Internet Connection Security Server is available from IBM. See the following Web pages for more information about the ICS:

<http://www.ics.raleigh.ibm.com/ics/icfgive.htm>

<http://www.ics.raleigh.ibm.com/icserver/>

<http://www.software.ibm.com/is/sw-servers/internet/cssdwn.html>

<http://www.software.ibm.com/is/sw-servers/>

- Lotus Domino Web Server Version 4.5.1 or higher.

For information about Lotus Web Servers visit:

<http://www.lotus.com>

On the Client

- Web browser software:
 - For OS/2 workstations:
 - Netscape Navigator, Version 2.02 or higher.
 - For non-OS/2 workstations:
 - Netscape Navigator, Version 3.01 or higher.
 - Microsoft Internet Explorer, Version 3.02 or higher.
- Other browsers may be used as long as they support frames and Java Script.

12.2 Required Hardware

On the Server

- Memory:
 - 1 megabyte of RAM. This is in addition to Communications Server and Web Server requirements.
- Disk:
 - 3 megabytes of disk space. This is in addition to Communications Server and Web Server requirements.

On the Client

- Video display:
 - The minimum required screen resolution is 640x480 pixels.
 - The minimum required color resolution is 256 colors.

12.3 Security

Use of the Web Server security mechanism should be implemented (but is not required). Most Web servers have instructions to add user ID and password authentication for specified directories. The Web Administration directory (\www\csc) and all subdirectories should have password protection. Unless a secure browser and secure server are used, the user ID and password will be transmitted in an encoded, but not encrypted, form over the network. While this is probably not a concern for an administrator using an internal network, unsecured usage over the open Internet could pose risks. For more information about secure browser and server software, visit:

<http://www.software.ibm.com/is/sw-servers/internet/>

12.4 Installation

The following installation is performed at the server machine. The only software you will be required to install in the client machine is a Web browser.

12.4.1 Installing the IBM ICSS 4.2.1

After downloading the file from the ICS Web page, you execute the following steps to install the Internet Connection Secure Server:

- Rename the file os2421use_enu.exe to OS2421.EXE
- Unpack: OS2421 C:\TEMP /D
- Install ICSS: C:\TEMP> INSTALL

Note

The IBM ICSS requires an HPFS formatted drive.

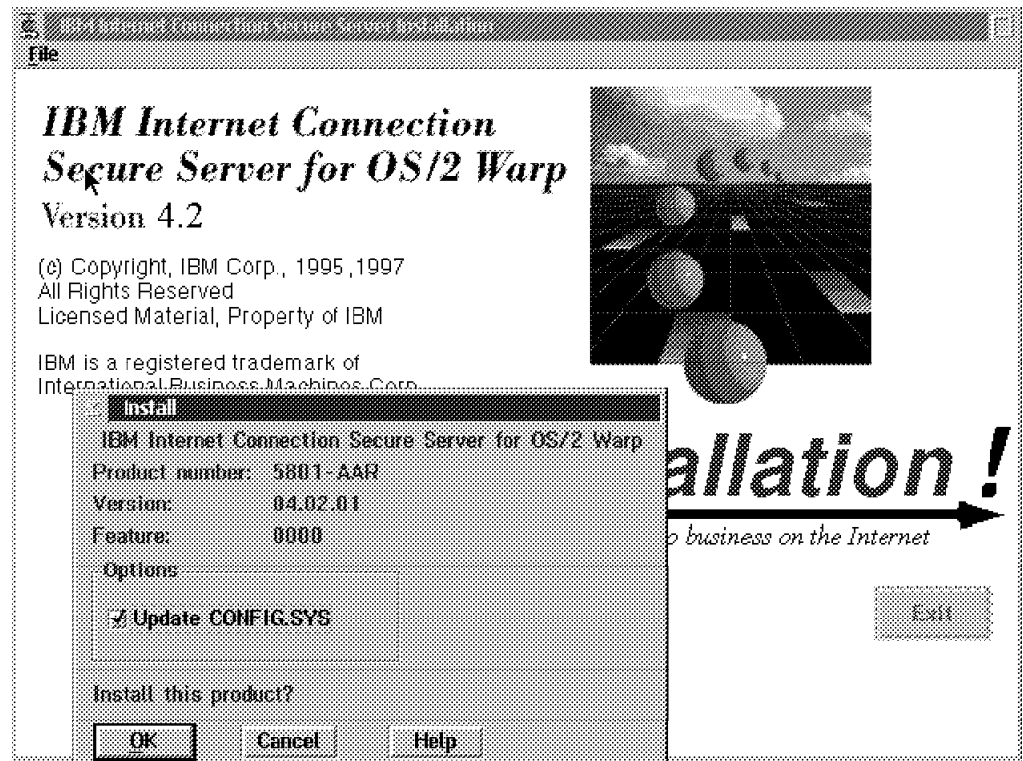


Figure 160. ICSS Install Page

Select **OK** to continue with the installation process.



Figure 161. ICSS Install - Directories

The IBM ICSS will be installed on WWW directory. Select **Install** to initiate the installation.

Internet Connection Secure Server Install Configuration

Enter host name, HTTP and SSL port numbers, and key ring file. Also enter the Administrator ID and password you want to use to protect the Configuration and Administration Forms.

Note: This Administrator ID and password will replace the existing one(s).

Host name Key ring file

HTTP port Administrator ID

SSL port Password

☒ Auto Start Server at Bootup

Target directories

Configuration file	C:\MPTM\ETC\httpd.cnf
CGI scripts	C:\WWW\CGI-BIN\
HTML documents	C:\WWW\HTML\
Configuration password file	C:\MPTM\ETC\ADMIN.PWD

Figure 162. ICSS Install Page

The Host name, Key ring file, ports, Administrator ID and Password are shown in Figure 162.

In our scenario, the administrator ID and password are used to access Web Administration from the Web browser.

Select **OK** to start the installation.

You will receive an information message if the installation was successful as shown in Figure 163 on page 244.

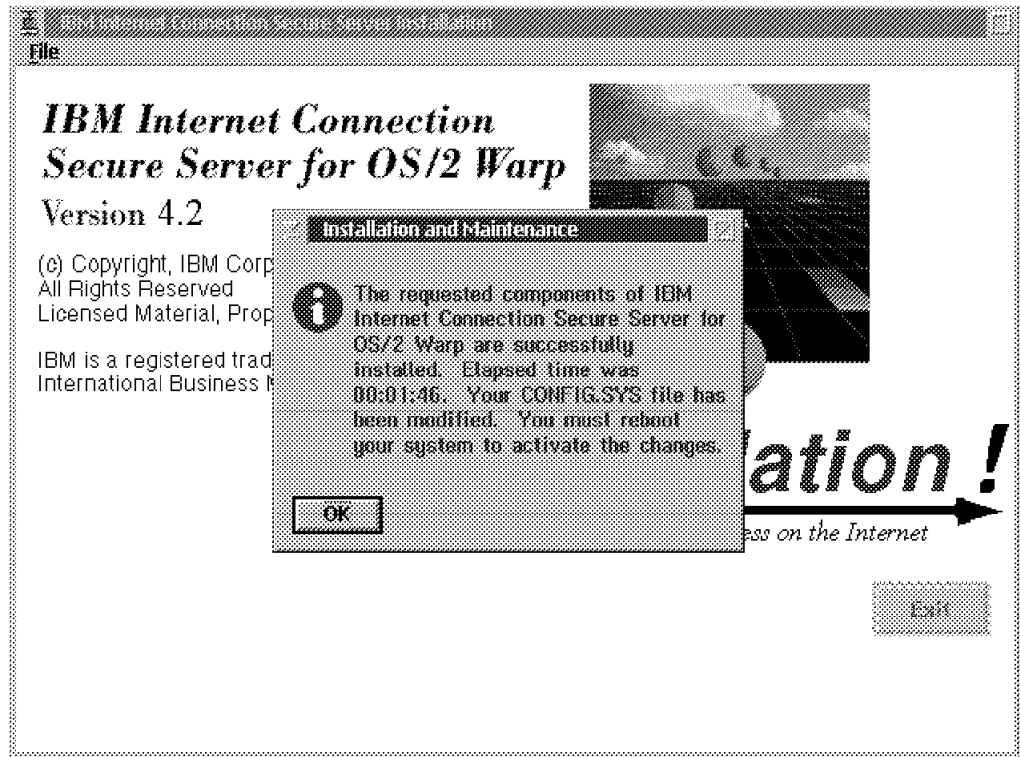


Figure 163. ICSS Install Page - Successful Installation

12.4.2 Installing Web Administration

After the ICSS installation, you will install Web Administration from the eNetwork Communications Server for OS/2 Warp CD. We invoke the installation option and select the Web Administration installation as shown in Figure 164 on page 245.

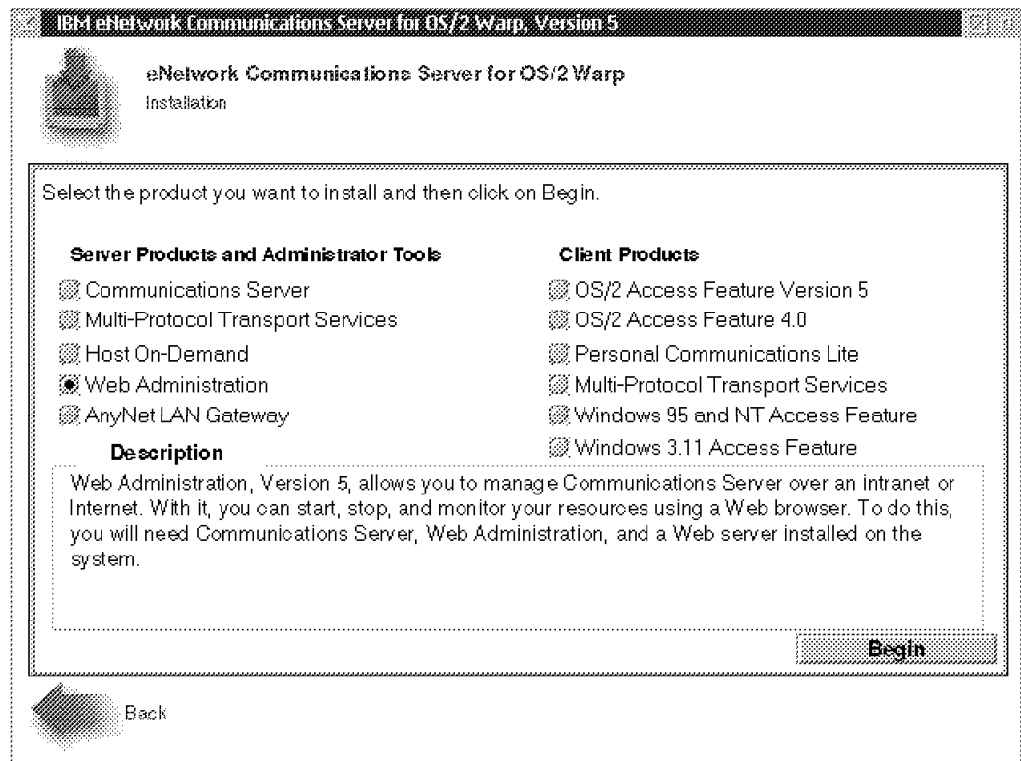


Figure 164. eNetwork Communications Server - Installation

In the Server Products and Administrator Tools choose **Web Administration** and click **Begin**.

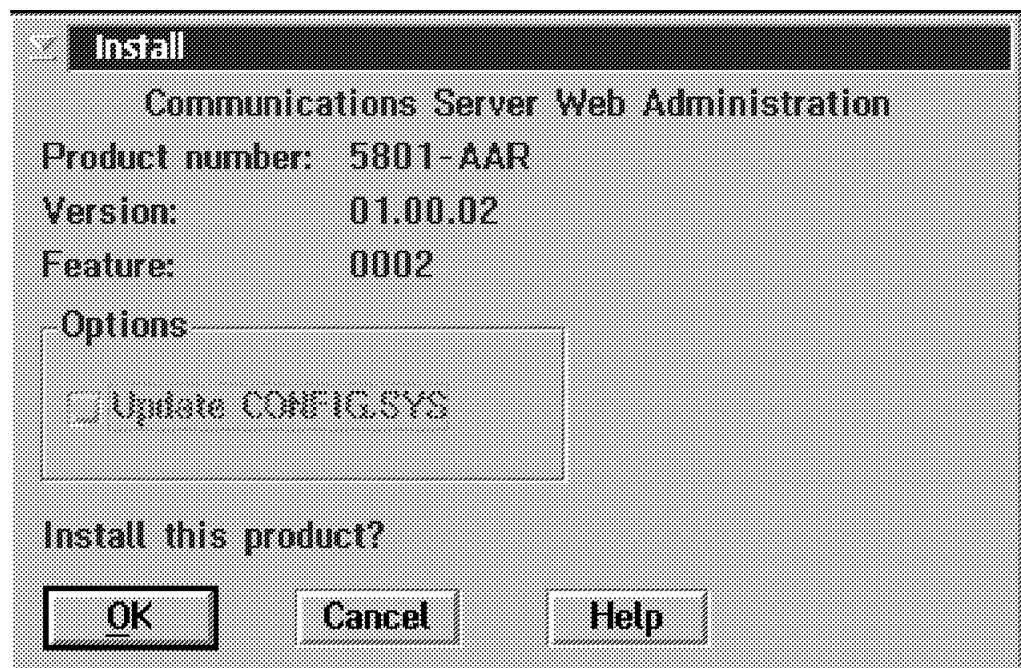


Figure 165. Web Administration Installation

Select **OK**.

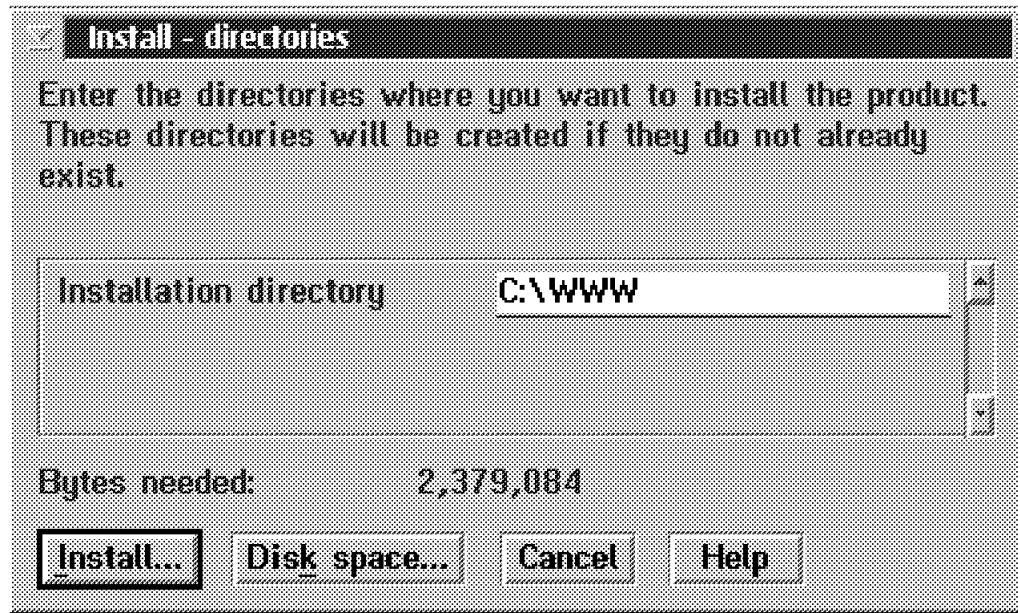


Figure 166. Web Administration Installation - Directories

The Web Administration will also be installed in the WWW directory.

Select **Install** to proceed with the installation.

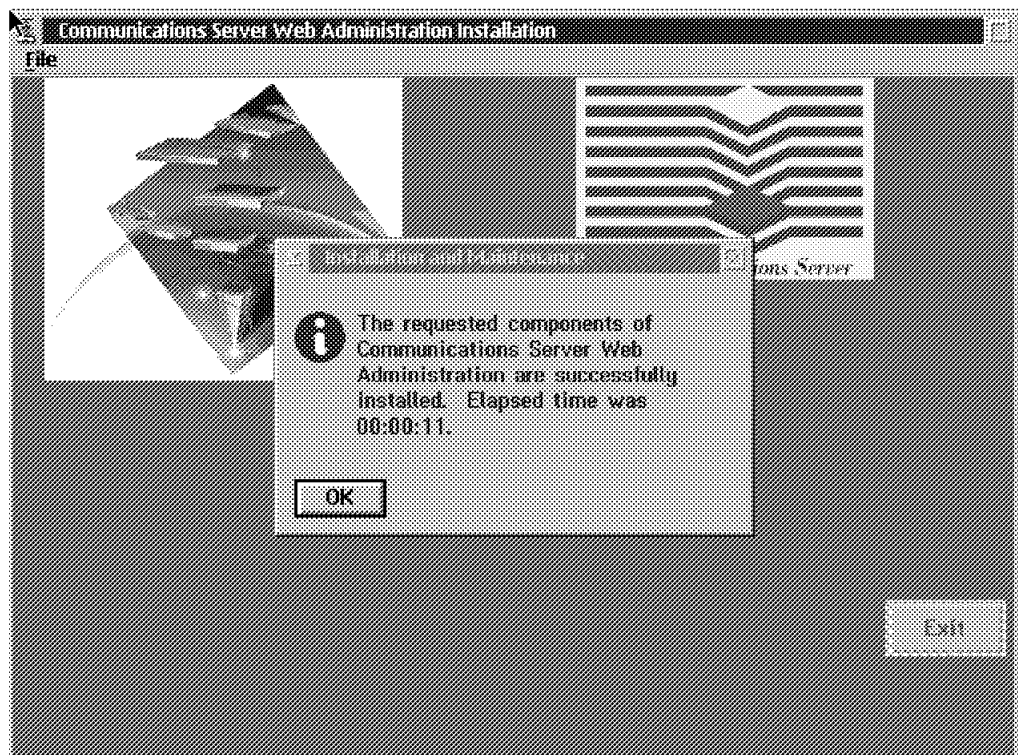


Figure 167. Web Administration - Successful Installation

You will receive an information message if the installation was successfully (see Figure 167).

12.5 Customizing the FFST/2

eNetwork Communications Server for OS/2 Warp uses the error logging facilities of FFST/2. On an unattended, remotely administered server, it is not desirable to allow FFST/2 to display informational or error messages in pop-up windows. To turn this function off, go to the FFST/2 folder and start the Message Console, or type EPWCONS from an OS/2 window. From the Message Console menu, select **Option - Configuration** and set Message pop-up to Off.

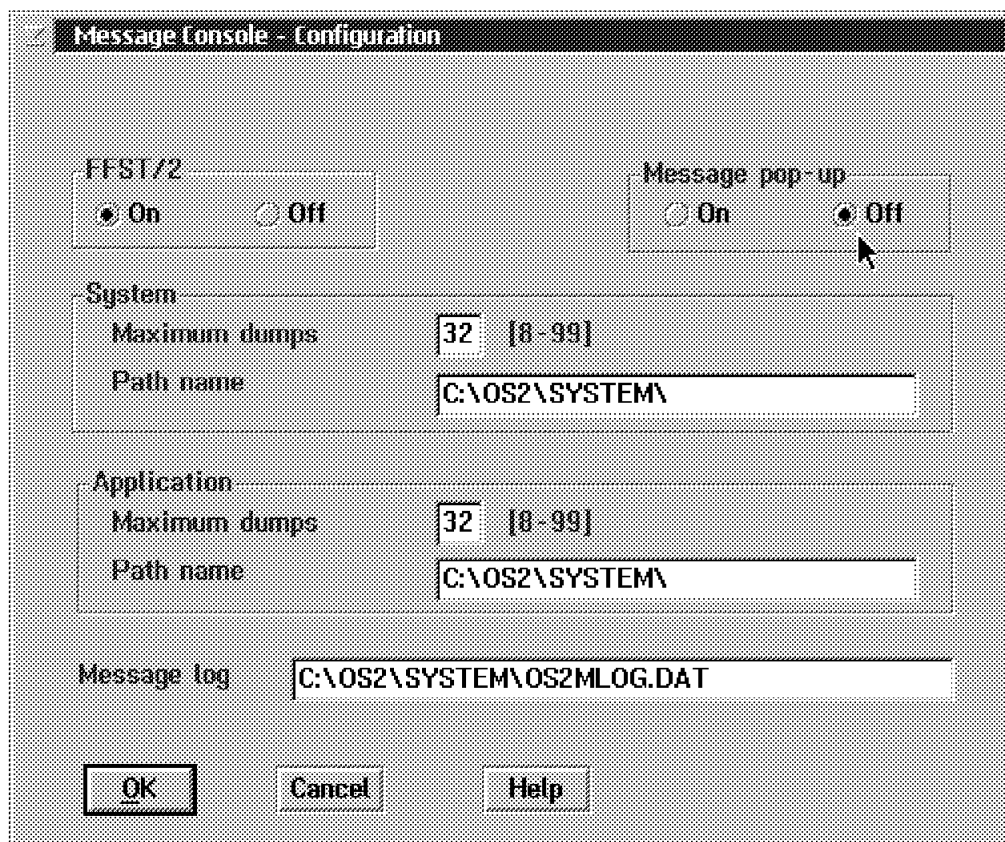


Figure 168. FFST/2 Configuration - Message Pop-Up Off

12.6 How to Use Web-Based Administration

In this section we show you how to log on and to use the Communications Server for OS/2 Web Administration.

After the Web Administration and required Web software have been installed and configured, you can begin managing Communications Server from a Web browser. Use your browser to access the following page:

`http://hostname/csc/main`

where **hostname** is the TCP/IP host name or IP address of the system running the Web server.

When you try the connection to Web server, the user ID and password are required.



Figure 169. Logging On

12.6.1 Web Administration Pages

In the Main page we have the navigation bar on the left. To perform a task, click on any of the choices in the navigation bar.

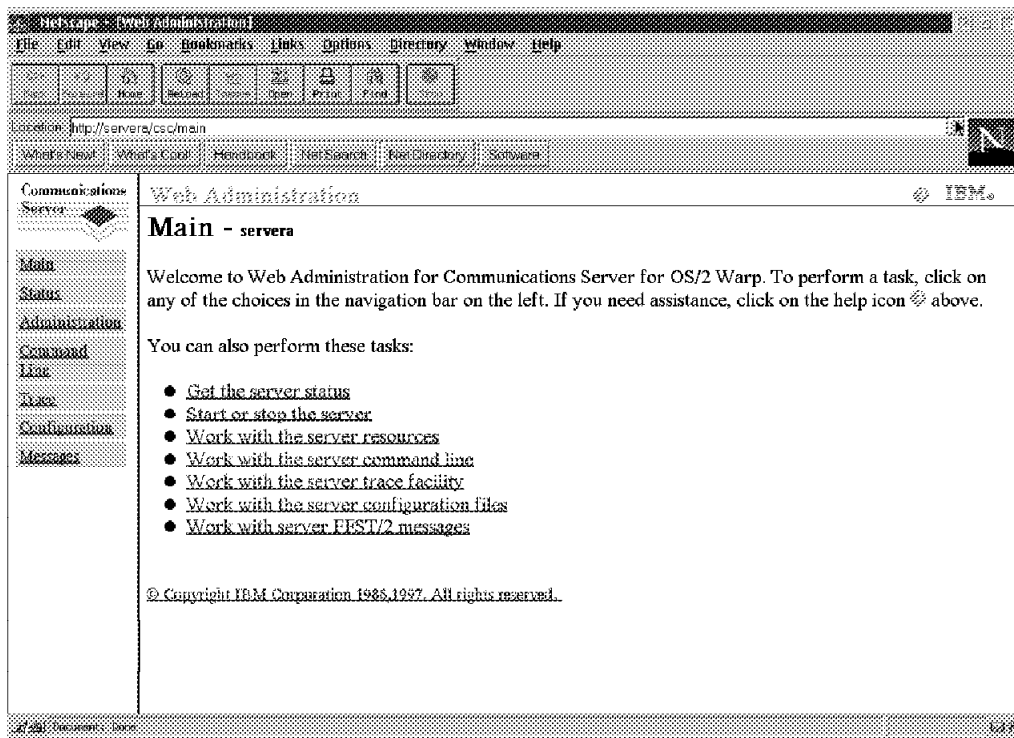


Figure 170. Main Page

Note

Do not use the browser's forward and back buttons. Use the Navigation bar.

If you choose the **Status** function, the connection's status from the Communications Server machine will be displayed.

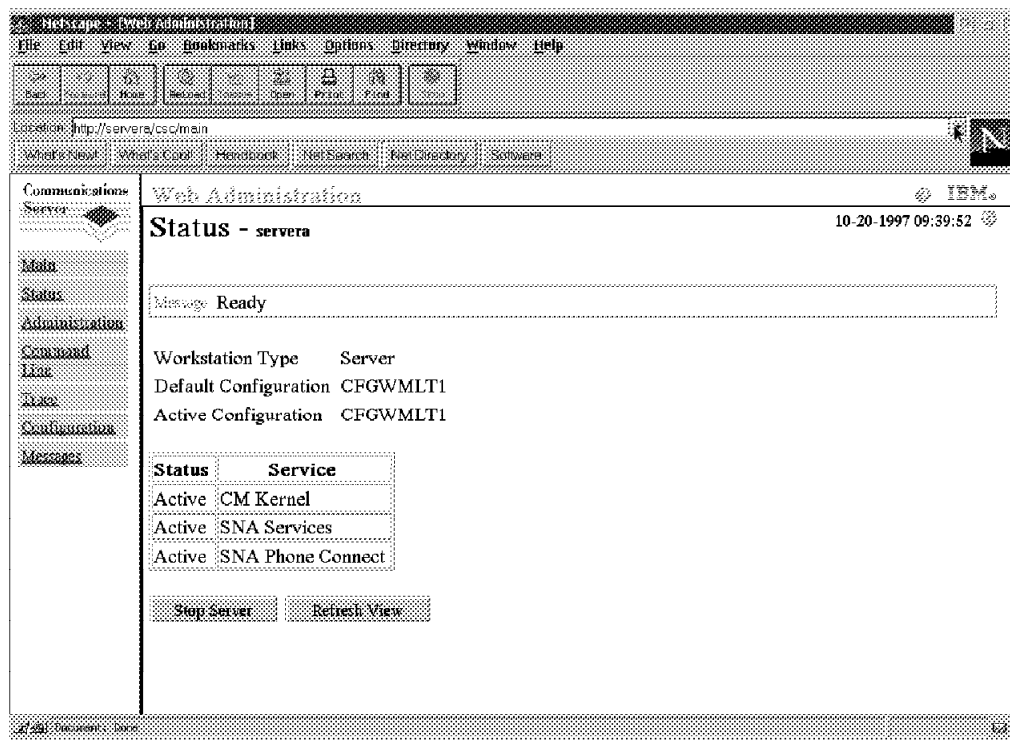


Figure 171. Connection Status

If you choose the **Administration** function, you can view a list of selectable resources.

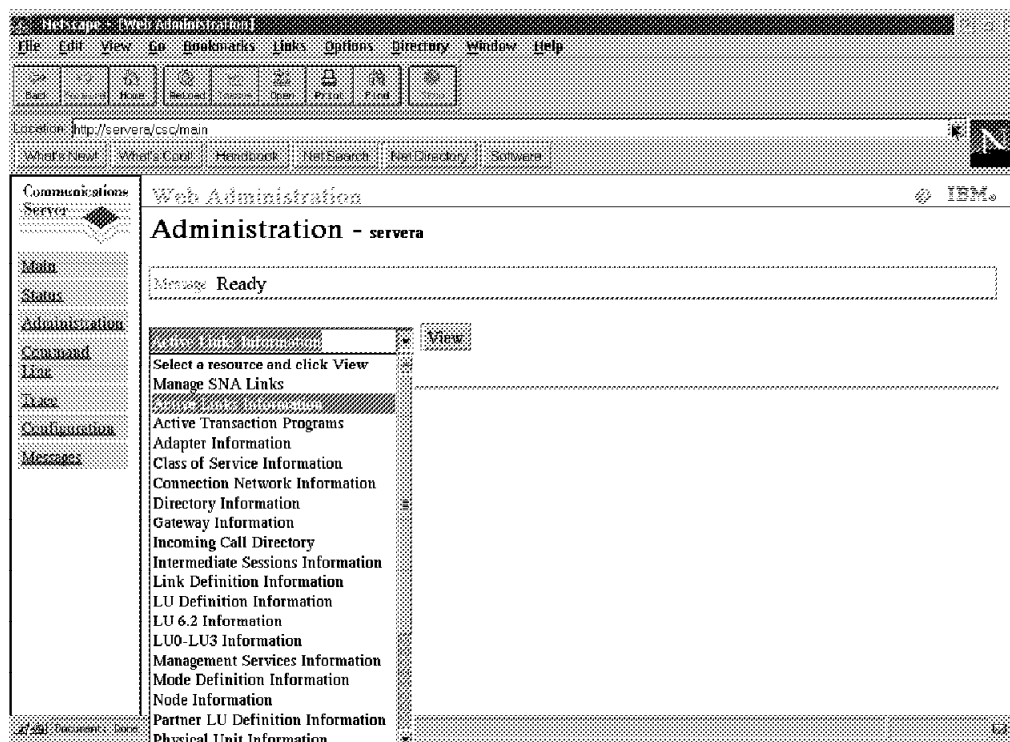


Figure 172. Administration Function

Selecting Manage SNA Links, will display the links where you can Activate, Soft deactivate or Hard deactivate those links.

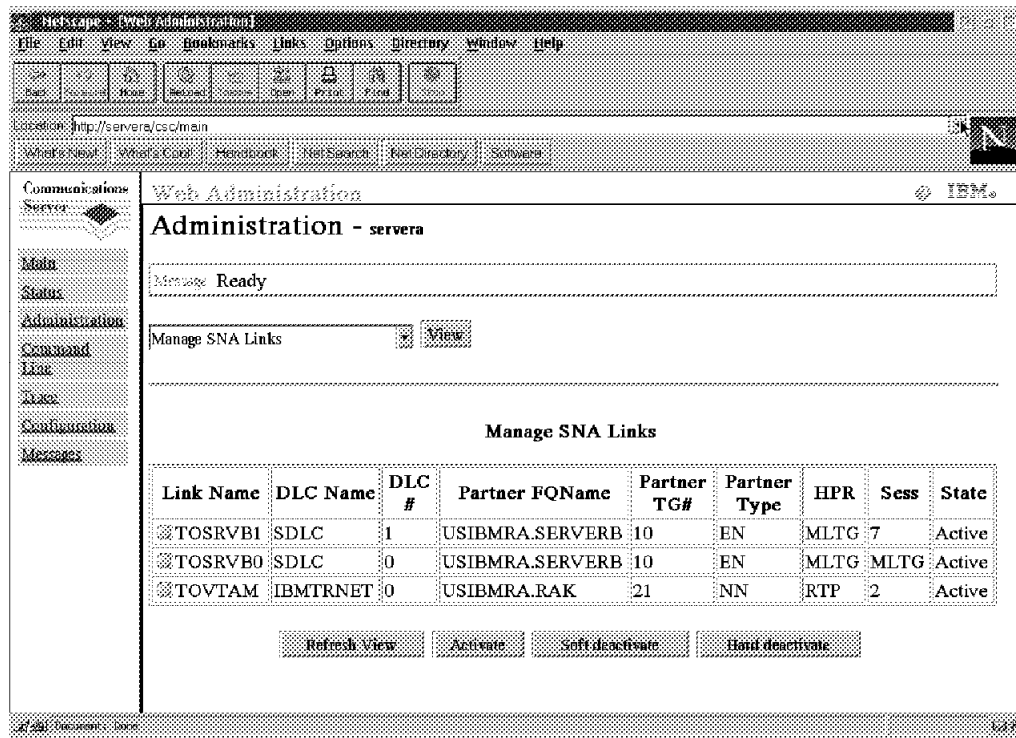


Figure 173. Administration Function

If you choose the **Command Line** function, you also can view a list of selectable resources as shown in Figure 174.

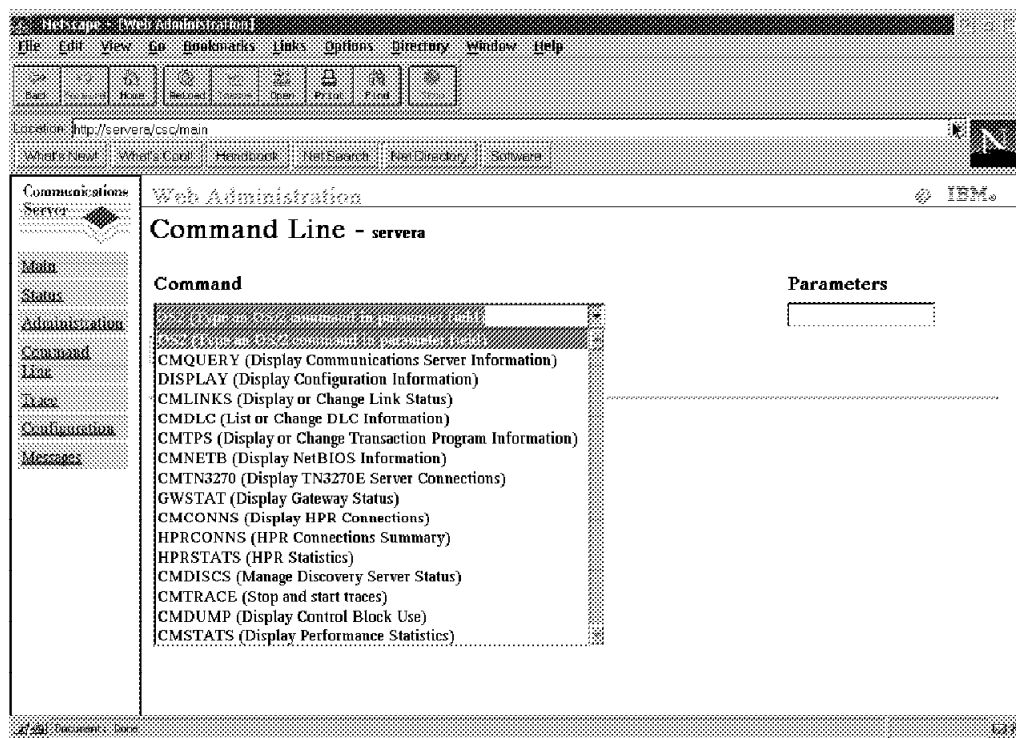


Figure 174. Command Line Function

You can choose **CMQUERY**, for example:

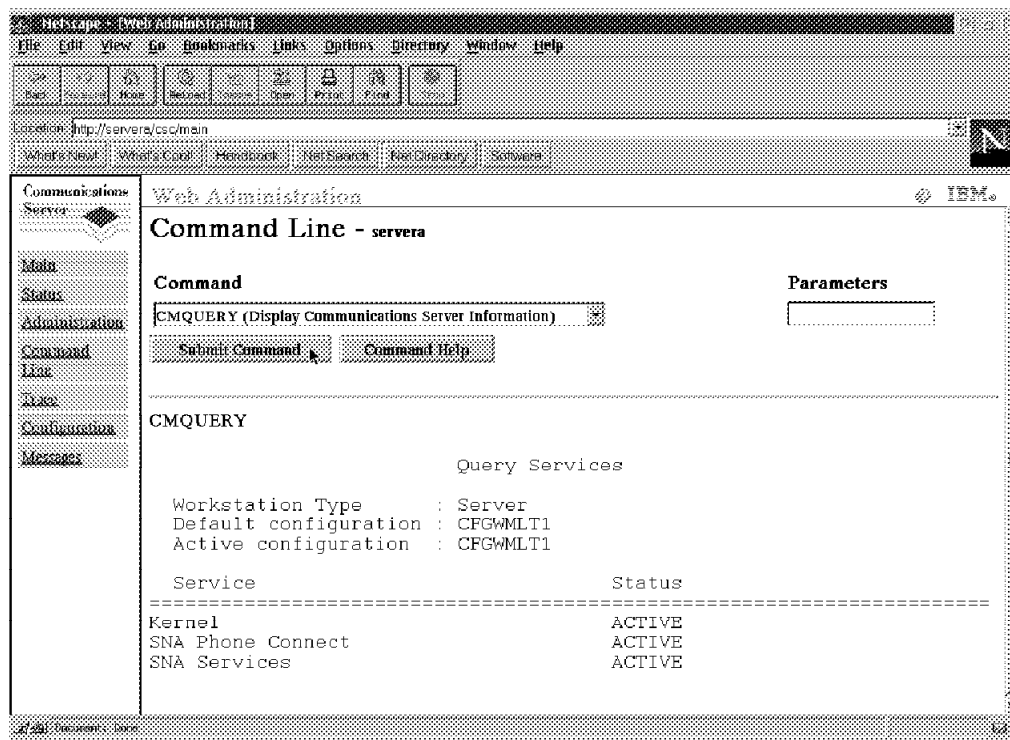


Figure 175. Command Line Function

Or you can choose **OS/2** (type an OS/2 command in Parameters field) and type **DIR** in the Parameters field. The result of the dir command will be displayed.

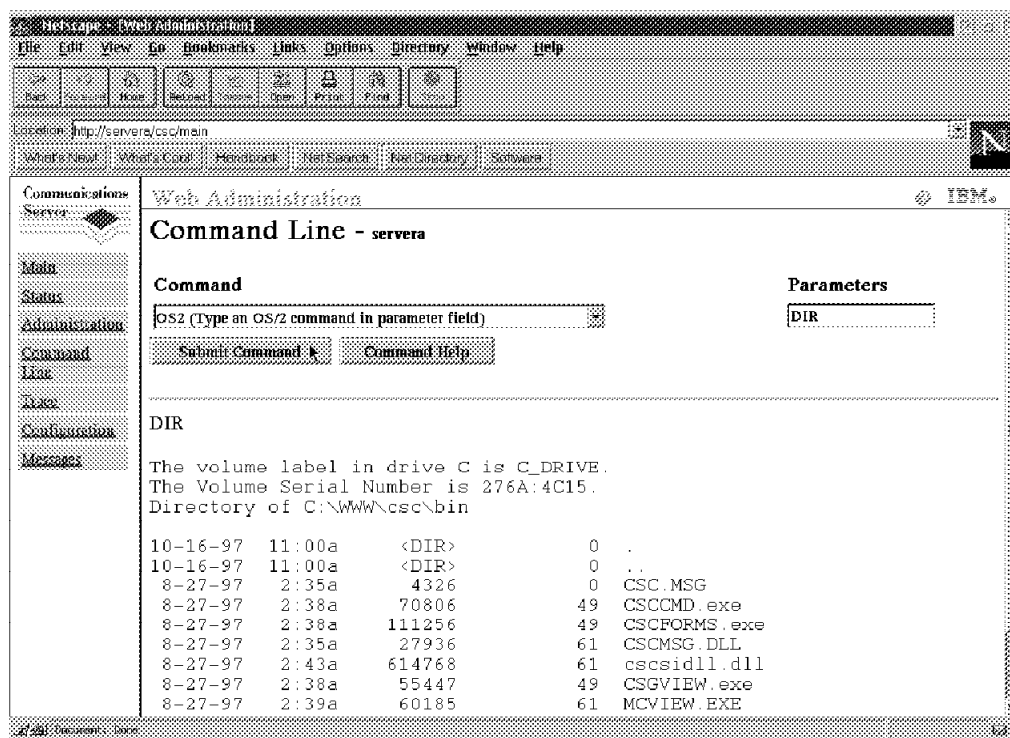


Figure 176. Command Line Function

Note

You cannot execute commands that call a graphic interface, for example, MPTS, CMSETUP, etc.

After a timeout of 30 seconds, you will receive a message that is shown in Figure 177 on page 252.

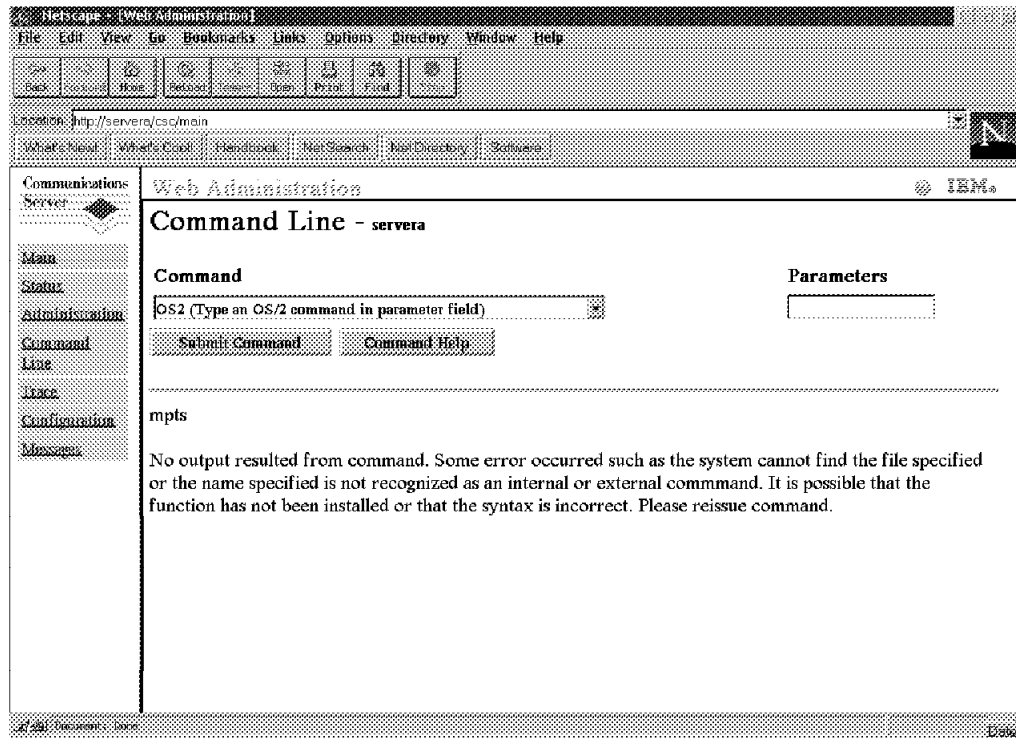


Figure 177. Command Line Function

With the **Trace** function you can start and view traces.

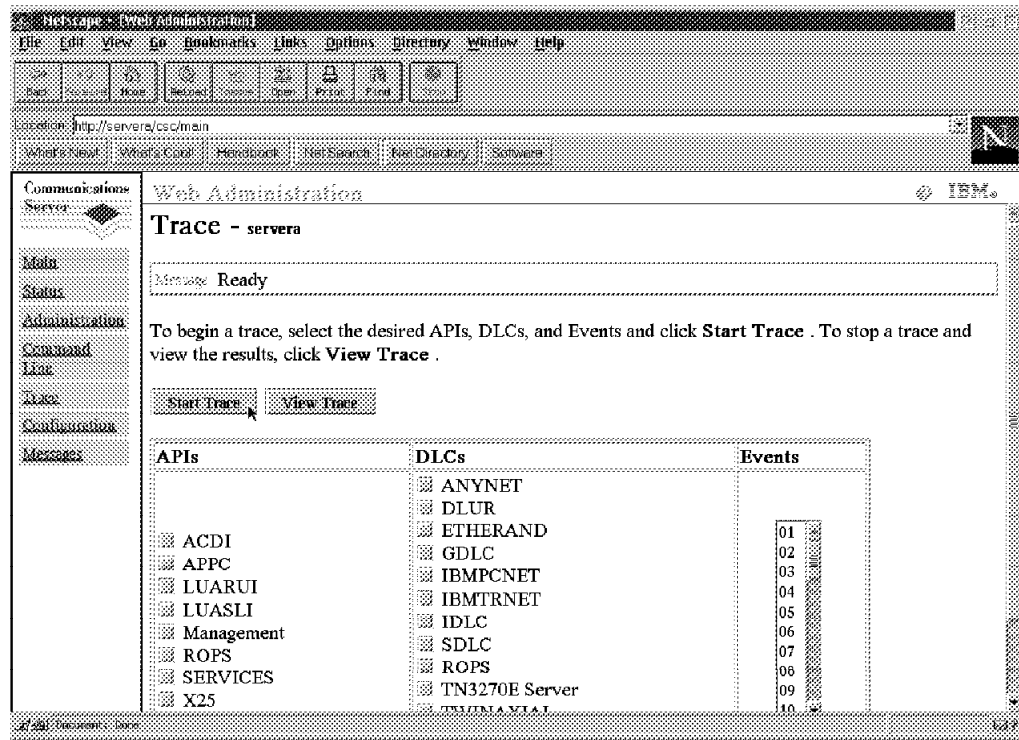


Figure 178. Trace Function

In the **Configuration** function you can view and modify the configuration files.

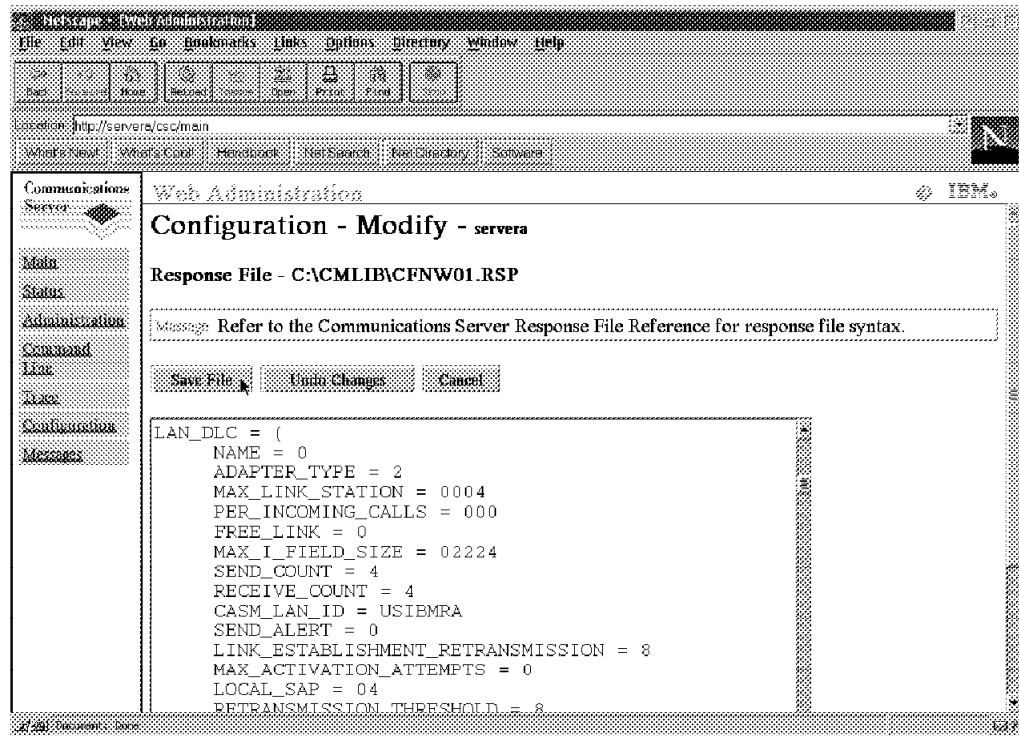


Figure 179. Configuration Function

In the **Messages** function you can view the message log from the Communications Server machine.

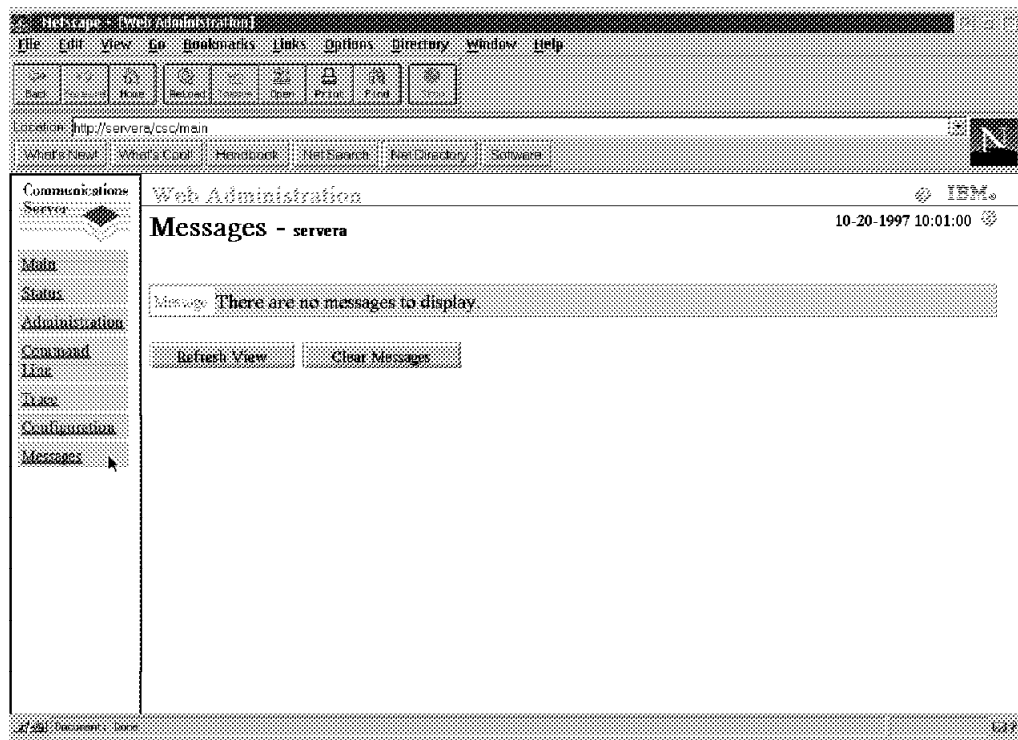


Figure 180. Messages Function

Part 8. Problem Determination

Chapter 13. Problem Determination

There are several options you can use for problem determination. Whether you are troubleshooting a configuration problem or if you suspect there is a defect, we recommend you follow the guidelines documented in this chapter.

In addition to the facilities we have from previous releases, eNetwork Communications Server for OS/2 Warp Version 5.0 has included support for ring 0 traces. It uses OS/2 features and several utility programs that will allow you to set, run and format ring 0 traces.

In general, eNetwork Communications Server for OS/2 Warp Version 5.0 modules run in ring 3 (application level) but there are certain components that need to run in ring 0. They are for example, device drivers, frame manager drivers and HPR ANR components among others.

Note

Ring 0 traces allow you to trace HPR ANR activity.

13.1 Troubleshooting Information

Some basic books for problem determination are SC31-8186-02 *IBM eNetwork Communications Server for OS/2 Warp Problem Determination Guide* and SG24-4916-00 *IBM Communications Server for OS/2 Warp Version 4.1 Enhancements*. But in eNetwork Communications Server for OS/2 Warp Version 5.0 there are new features.

For example, for the new branch extender support we have:

- Two new fields on the DEFINE_LOCAL_CP verb
- A new field on the DEF_ADAPTER and the DEF_LINK verbs
- A new field in the XID3 denoting BrEx downlink indicator
- A new bit in the CV'4680'x denoting BrEx uplink TG

In the online help you have a good explanation of the problem determination in a step-by-step manner.

You can also find online help on the Internet:

<http://www.networking.ibm.com/cms/pubcs250/qbhtml.html#HDRPD>

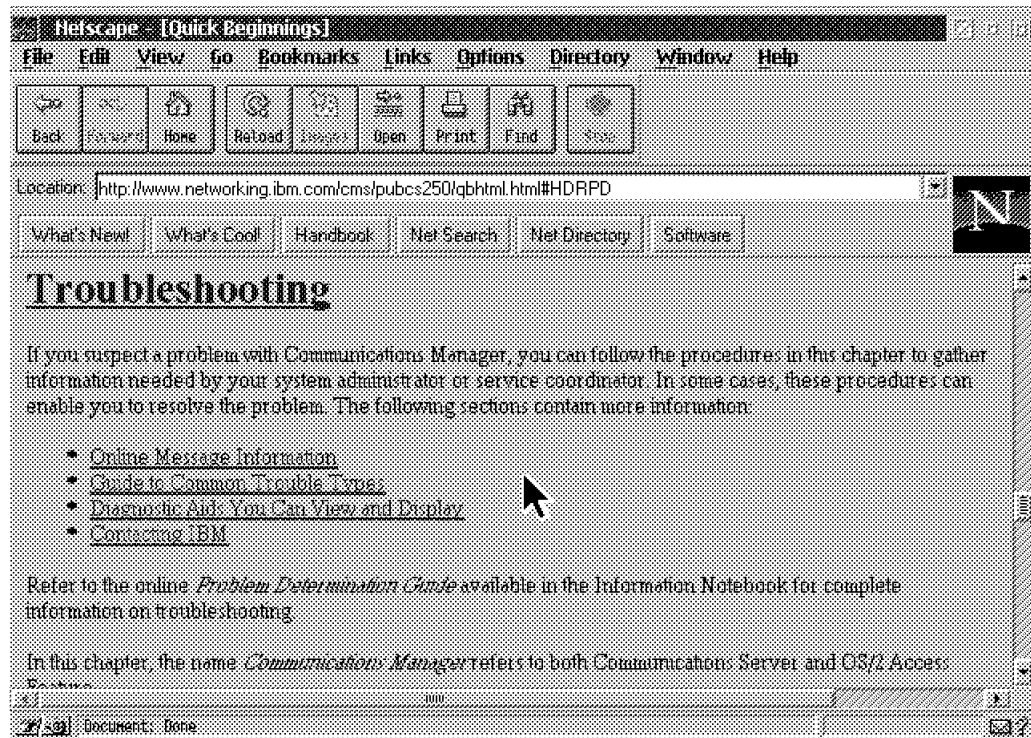


Figure 181. Problem Determination - Online Help

13.2 FFST/2

FFST/2 is a software problem determination tool for OS/2 system software and applications. FFST/2 is designed to capture error data when the error occurs. It provides immediate problem notification to predefined locations, and furnish a unique error code identification. Because FFST/2 remains inactive until a software error is detected, impact on system performance is minimal.

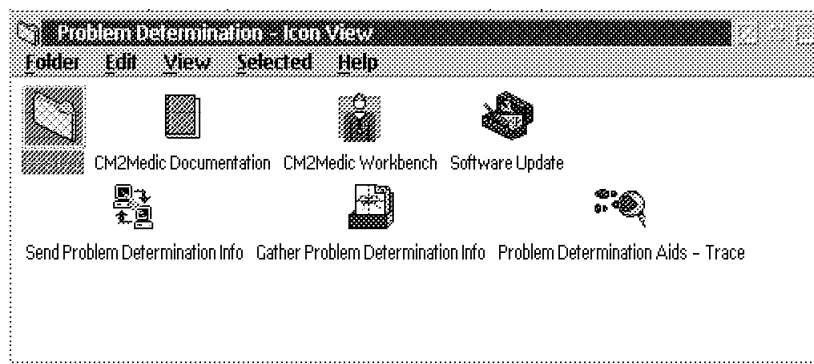


Figure 182. Problem Determination Folder

You should always invoke FFST/2 when troubleshooting a configuration problem. Most of the information provided by FFST/2 will help you to isolate an error situation. The services FFST/2 provides include:

- Pop-up messages
- A message log formatter

- Access to the OS/2 system error log
- A dump formatter
- A message console
- A command line interface for FFST/2 initialization and configuration

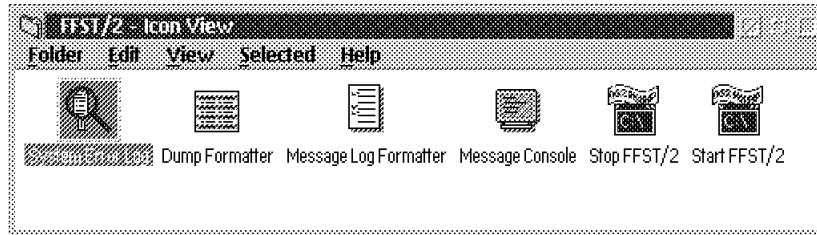


Figure 183. FFST/2 Folder

13.3 APPNT and APPNF

If obtaining traces is your last resort, there are two commands you can use from the command line:

- APPNT
 - Traces APPC, T/R, HPR
 - Invokes CMTRACE by executing the following command:
`C:\CMTRACE START -reset -api appc services management -data
 ibmtrnet hpr -event 1 2 3 4 5 12 36 38 -storage 16`
- APPNF <filename>
 - Stops traces
 - Copies traces to a file
 - Obtains summary and detailed reports
 - APPNF <filename> generates:
 - C:\CMTRACE STOP
 - C:\CMTRACE copy <filename>.trc
 - C:\FMTTRACE <filename>.trc +d +s

We now include a listing of the APPNT command provided by eNetwork Communications Server for OS/2 Warp in order to illustrate how APPNT invokes CMTRACE to start traces and what options are used (see Figure 184 on page 260).

```

@ECHO OFF
REM ** This sample command file will reset the trace buffer and start
REM ** traces for APPC and Services verbs, Token Ring and HPR Data and
REM ** APPC events 1 2 3 4 5 and 12 using the CMTRACE program.  CMTRACE
REM ** is invoked with the following parameters:
REM **
REM ** CMTRACE START -reset -api appc services management -data ibmtrnet hpr -event 1 2 3 4 5 12
REM **                  -storage 16

REM **
REM ** -storage 16 uses up to sixteen 64K segments (1 megabyte) for traces
REM ** -trunc 600 truncates records at 600 bytes to reduce trace storage used
REM **
REM ** For other options in tracing enter "CMTRACE ?"
REM **
REM ** The following is a complete list of system event numbers and their
REM ** corresponding trace selections:
REM **
REM **   System Event      Trace Choice      Description
REM **   -----
REM **   01                APPC_CCB          internal DLC interface
REM **   02                APPC_INT          internal APPC flows
REM **   03                APPC_PROCESS      internal SNA process dispatch
REM **   04                APPC_SNDRCV       internal APPC send and receive
REM **   05                APPC_XID          XID connection flows
REM **   06                ACDI
REM **   10                IBMPCNET
REM **   11                SDLC
REM **   12                SERVICES_INT      utility verbs (errorlog & convert)
REM **   14                IBMTRNET
REM **   19                X25_API EVT_1
REM **   20                X25_DLC
REM **   21                TWINAXIAL
REM **   23                X25_API INTERNAL
REM **   24                X25_API QTHREAD
REM **   25                LUA
REM **   29                ETHERAND
REM **   30                SUBSYSM
REM **   31                APPC_LOCAL_PATH
REM **   32                AnyNet Internal
REM **   33                MS Support for Netware
REM **   34                DLUR
REM **   35                Discovery
REM **   36                HPR RTP Header
REM **   37                HPR RTP Payload
REM **   38                HPR Internal
REM **   39                HPR RTP Flow Control
REM **   40                HPR Module
REM **   41                TN3270E Server Internal
REM **
ECHO ON
CMTRACE START -reset -api appc services management -data ibmtrnet hpr -event 1 2 3 4 5 12 36 38
                  -storage 16

```

Figure 184. APPNT Command

13.4 Trace Services

Problem determination aids in eNetwork Communications Server for OS/2 Warp offer trace services where you can select the type of trace entries in three categories (see Figure 185):

- APIs
- Data Link Controls (DLCs)
- Events

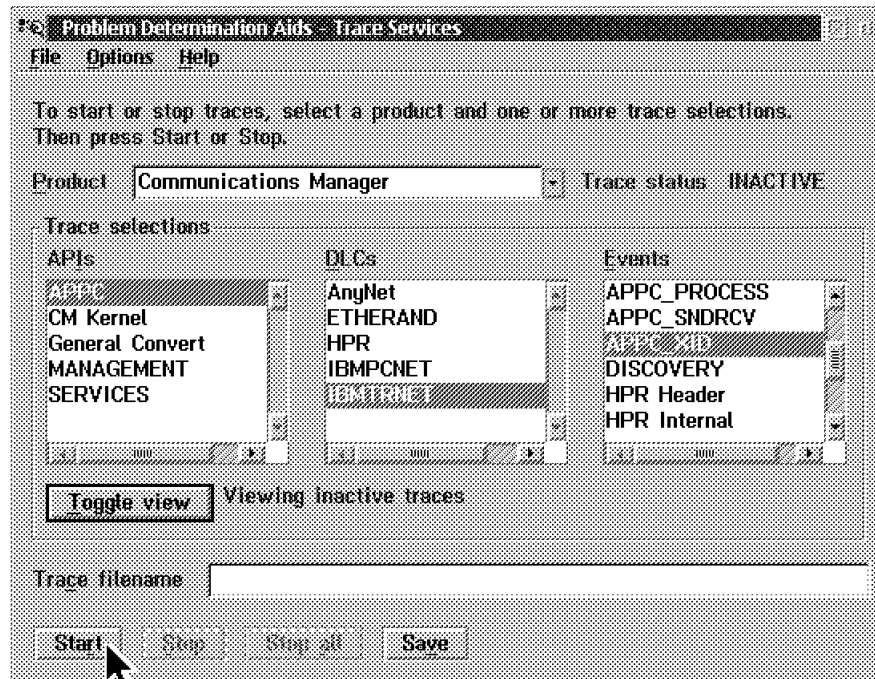


Figure 185. Starting Trace Services

After running the scenario you are tracing you will need to stop and save the trace activity to a file (see Figure 186 on page 262).

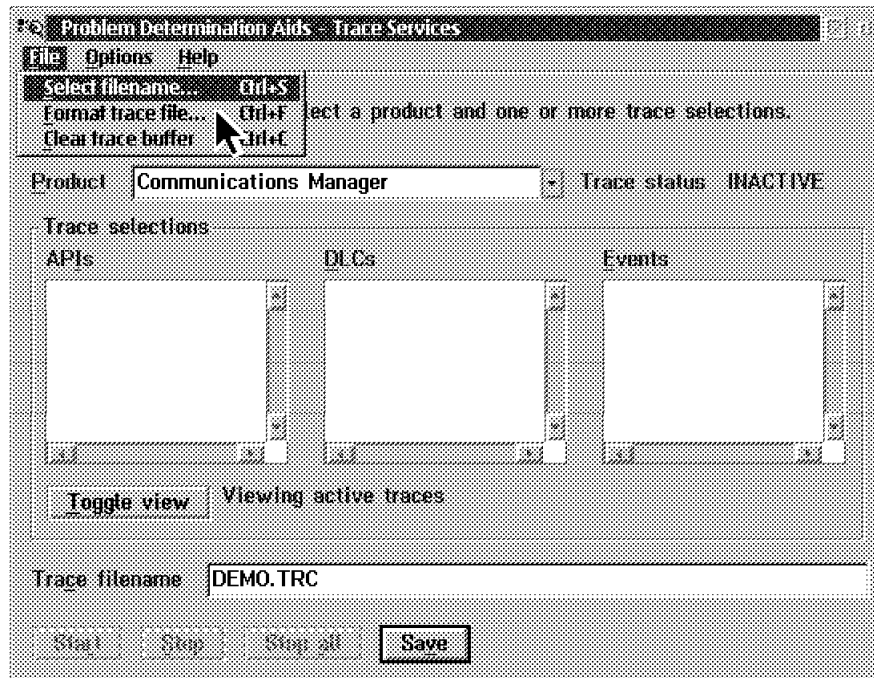


Figure 186. Saving Traces

The result is a raw trace file which is readable but not interpreted (see Figure 187).

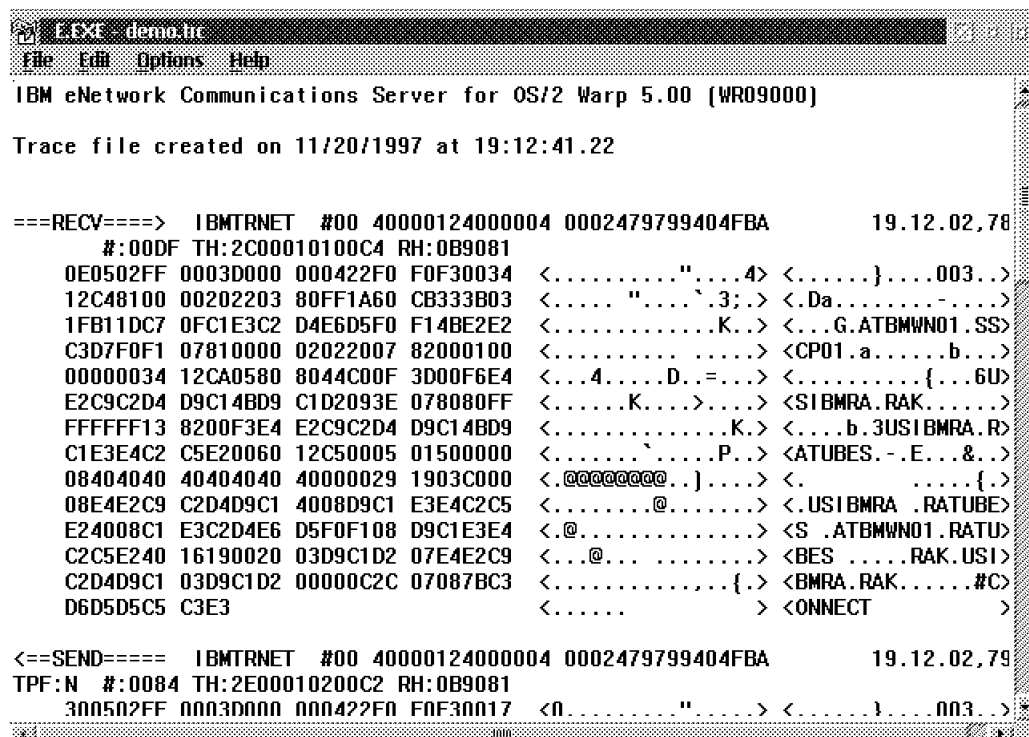
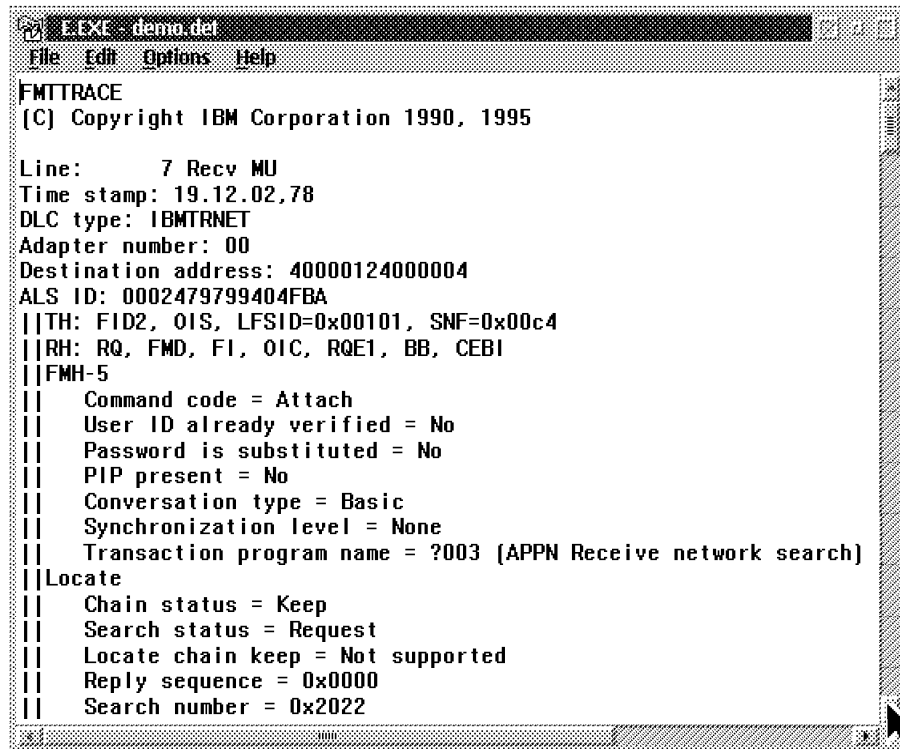


Figure 187. Unformatted Trace File

You will then select the format trace file option to format the raw trace and obtain a summary file and a detailed trace file (see Figure 188 on page 263).



```
demo.dcl
File Edit Options Help
FMTTRACE
(C) Copyright IBM Corporation 1990, 1995

Line:      7 Recv MU
Time stamp: 19.12.02,78
DLC type:  IBMTRNET
Adapter number: 00
Destination address: 40000124000004
ALS ID: 0002479799404FBA
||TH: FID2, OIS, LFSID=0x00101, SNF=0x00c4
||RH: RQ, FMD, FI, OIC, RQE1, BB, CEBI
||FMH-5
||  Command code = Attach
||  User ID already verified = No
||  Password is substituted = No
||  PIP present = No
||  Conversation type = Basic
||  Synchronization level = None
||  Transaction program name = ?003 (APPN Receive network search)
||Locate
||  Chain status = Keep
||  Search status = Request
||  Locate chain keep = Not supported
||  Reply sequence = 0x0000
||  Search number = 0x2022
```

Figure 188. Formatted Trace File

13.5 Ring 0 - Traces

eNetwork Communications Server for OS/2 Warp Version 5.0 has included support for ring 0 traces. This facility will allow you to trace drivers and other components that run in ring 0 in your system. Specifically you can use this facility to obtain HPR ANR traces if required.

13.5.1 Setting Ring 0 Traces

Since this support uses the OS/2 function to obtain ring 0 traces you will be required to set some options in the config.sys file of your system and you will be required to reboot your machine for the options to take place.

The following options are required in config.sys in order to set ring 0 traces in your system:

- TRACEBUF=63
 - Required by OS/2 for ring 0 tracing
 - Uses 64 KB buffer
- DEVICE=C:\CMLIB\ACSHPRDD.SYS /T
 - /T parameter required by ANR device driver

The following commands are available to set traces on and off for HPR ANR or any other ring 0 traces:

- TRACE ON 144 /C
 - Starts OS/2 ring 0 traces for major code 144

- TRACE OFF 144
 - Stops ring 0 traces
- TRACEFMT
 - Starts PM program which captures ring 0 trace buffer and allows traces to be saved in a file (for example, TRACE1.FTF) as shown in Figure 189.

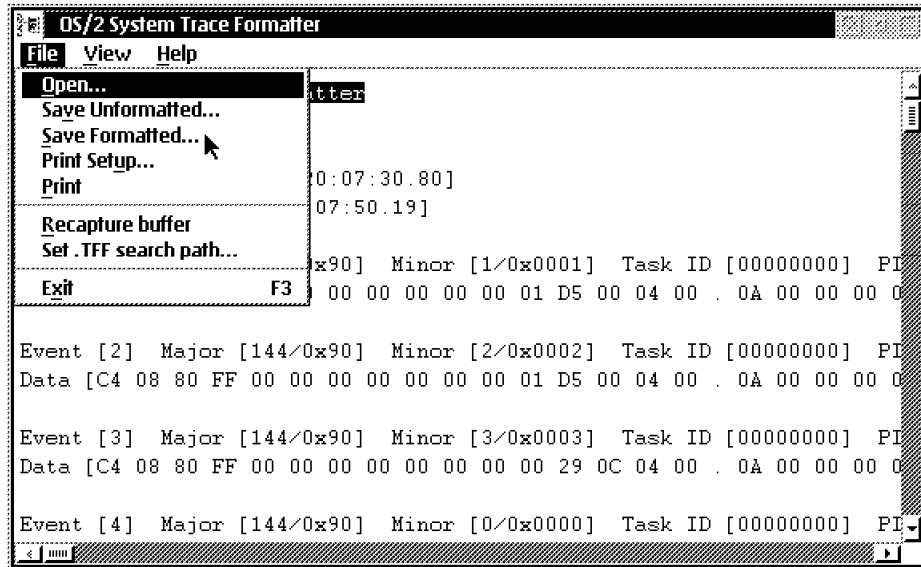


Figure 189. Saving Unformatted or Formatted Ring 0 Traces

13.5.2 Formatted ANR Traces

In addition to these commands, there is a program (ANRTSUM) that allows you to format traces with ANR entries. For example if the name of the trace file is trace1.ftf and you want to format HPR ANR traces to file trace1.anr, you will enter the following command:

```
ANRTSUM TRACE1.FTF > TRACE1.ANR
```

A sample ANR formatted trace is shown in Figure 190 on page 265.

EVENT#	TCID	TRACEPOINT	ANR LABEL SET	PID	PRIORITY	DOFF	DATALEN	BYTESEQ#	TIME
000001	01d5	ToRtp	80ff	0000	High	0028	00000000	000001c5	[20:08:12.33]
000002	01d5	NlpFromldc	80ff	0000	High	0028	00000000	000001c5	[20:08:12.33]
000003	0029	NlpToIdc	80ff	003b	High	0028	00000000	00000232	[20:08:12.33]
000004	0029	FromRtp	800280ff	003b	High	0028	00000000	00000232	[20:08:12.33]
000005	01d4	ToRtp	80ff	0000	Net	0028	00000000	000000d1	[20:08:10.11]
000006	01d4	NlpFromldc	80ff	0000	Net	0028	00000000	000000d1	[20:08:10.11]
000007		Send_MU		003b	040c479340bd48b9ac31832c	0400			[20:08:10.11]
000008	0022	NlpToIdc	80ff	003b	Net	0028	00000000	00000127	[20:08:10.11]
000009	0022	FromRtp	800280ff	003b	Net	0028	00000000	00000127	[20:08:10.11]
000010	0030	NlpToIdc	80ff	003b	Net	0028	00000000	000005e4	[20:08:07.09]
000011	0030	FromRtp	800880ff	003b	Net	0028	00000000	000005e4	[20:08:07.09]
000012	01d1	ToRtp	80ff	0000	Net	0028	00000000	00000aae	[20:08:07.09]
000013	01d1	NlpFromldc	80ff	0000	Net	0028	00000000	00000aae	[20:08:07.09]
000014	0035	NlpToIdc	80ff	003b	High	0028	00000000	000002b9	[20:07:40.90]
000015	0035	FromRtp	800880ff	003b	High	0028	00000000	000002b9	[20:07:40.90]
000016	01cd	ToRtp	80ff	0000	High	0028	00000000	0000036e	[20:07:40.90]
000017	01cd	NlpFromldc	80ff	0000	High	0028	00000000	0000036e	[20:07:40.90]
000018	0033	NlpToIdc	80ff	0000	High	0034	00000000	00000973	[20:07:34.20]

Figure 190. ANR Formatted Traces

13.5.3 Trace Spooling

When running ring 0 traces, the buffer used to store the trace entries is only 64 KB. Because of this it wraps easily and you can end up losing valuable data. In order to circumvent this situation, OS/2 Version 4.x solves this problem by using a new program called TRSPOOL.EXE. This executable is now shipped with OS/2. TRSPOOL periodically copies the trace buffer to multiple files and therefore you can get a complete trace of scenarios that create many entries in the 64 KB trace buffer.

13.5.3.1 ANRTSTRT.CMD

ANRTSTRT.CMD is an eNetwork Communications Server for OS/2 Warp command that invokes OS/2 TRSPOOL.EXE with proper parameters. The syntax of the ANRTSTRT command is as follows:

ANRTSTRT <trace duration (msecs)> [/s <trace in each file (msecs)>]

For example, you can run the following command in the directory where the spool files will be saved:

ANRTSTRT 30000 /s 500

In this command the following values are assumed:

- Default is 2000 msecs of trace to each file.
- Spool files are 64 KB each and they are created with names tracebuf.000, tracebuf.001,... and so on. These files can be viewed and formatted using the TRACEFMT.EXE program.
- In this case, the space required is approximately = $30000 / 500 * 64K = 3.9$ MB.

13.5.3.2 ANRTSTOP.CMD

The ANRTSTOP command is used to stop the TRSPOOL.EXE. It uses KILL.EXE and GREP.EXE. Note also that the TRACE STOP command does not work with spooling.

Note

ANRTSTRT and ANRTSTOP are not shipped with eNetwork Communications Server for OS/2 Warp. You should contact your IBM representative if you need this support.

Appendix A. TCP62 API Sample Program

In this appendix we show you a sample program that creates or updates an existing configuration in order to support SNA over TCP/IP.

A.1 TCP62 Sample Program - Header File

```
/**INC+*****  
/* Header:    tn62api.h                                */  
/*                                                    */  
/* Purpose:   Defines types and macros used by the TN62 configuration API */  
/*                                                    */  
/* Dependencies:                                     */  
/*                                                    */  
/* Invocation:                                       */  
/*           #include <tn62api.h>                     */  
/*                                                    */  
/* Copyright Notice:                                 */  
/*           IBM Personal Communication Version x.x for Windows xx.    */  
/*****@CM2CPYR*/  
/*           OCO SOURCE MATERIALS                        @CM2CPYR*/  
/*           IBM CONFIDENTIAL                            @CM2CPYR*/  
/*           (IBM Confidential-Restricted when combined with @CM2CPYR*/  
/*           the aggregated OCO source modules for this program) @CM2CPYR*/  
/*                                                    @CM2CPYR*/  
/*           5622-078                                    @CM2CPYR*/  
/*           (C) Copyright IBM Corp. 1988, 1997          @CM2CPYR*/  
/*                                                    @CM2CPYR*/  
/*           The source code for this program is not published or @CM2CPYR*/  
/*           otherwise divested of its trade secrets, irrespective of @CM2CPYR*/  
/*           what has been deposited with the U.S. Copyright Office. @CM2CPYR*/  
/*****@CM2CPYR*/  
/*           ALL RIGHTS RESERVED                          */  
/*                                                    */  
/**INC-*****  
  
#ifndef _TN62API_H_  
#define _TN62API_H_  
  
/*****  
/* Include base NOF API headers                                */  
/*****  
#include <appc_c.h>  
#include <appccfg.h>  
  
/*****  
/* Define this header file to be C if this is included by a C++ source file */  
/*****  
#ifdef __cplusplus  
extern "C" {  
#endif  
  
/*****  
/* For Win32 environment:                                     */  
/* - ensure correct packing                                   */  
/* - remove references to far                                 */
```

```

/*****
#pragma pack(4)
#define far

/*****
/* TN62 Operation Codes */
/*****
#define AP_START_TN62          0x2201
#define AP_STOP_TN62          0x2202
#define AP_DEFINE_LOCAL_LU_TN62 0x2221
#define AP_DEFINE_PARTNER_LU_TN62 0x2237

/*****
/* TN62 Primary Return Codes */
/*****
#define TN62_ERROR            0x9000

/*****
/* TN62 Secondary Return Codes */
/*****
#define TN62_NODE_RUNNING_NO_ANYNET 0x90000001L
#define TN62_CONFIGURATION_FILE_ERROR 0x90000002L
#define TN62_NODE_NOT_STARTED 0x90000003L
#define TN62_NODE_START_INCOMPLETE 0x90000004L
#define TN62_PARAMETERS_NOT_USED 0x90000005L
#define TN62_NAME_GENERATION_ERROR 0x90000006L

/*****
/* Miscellaneous TN62 defines */
/*****
#define TN62_MAX_HNAME        220

/**STRUCT+*****/
/* Structure: START_TN62 */
/*
/* Description: START_TN62 verb control block
/*
/*****

typedef struct start_tn62
{
    USHORT opcode; /* AP_START_TN62 */
    UCHAR reserv1[6]; /* Reserved */
    USHORT primary_rc; /* Primary RETURN_CODE */
    UCHAR reserv2[2]; /* Reserved */
    ULONG secondary_rc; /* Secondary RETURN_CODE */
    UCHAR reserv3[4]; /* Reserved */
    UCHAR key[8]; /* Key (ASCII) */
    UCHAR fqcp_name[17];
    UCHAR cp_alias[8];
    UCHAR update_cfg_file;
    UCHAR reserv4[2]; /* Reserved */
    ULONG ip_address_mask;
    USHORT connection_retry_secs;
    USHORT unacked_dg_retry_secs;
    USHORT unsent_dg_retry_secs;
    USHORT inactivity_timer_secs;
    USHORT conwait_secs;
    UCHAR domain_name_suffix[TN62_MAX_HNAME];
} START_TN62;

```

```

/**STRUCT-*****/

/**STRUCT+*****/
/* Structure: STOP_TN62 */
/* */
/* Description: STOP_TN62 verb control block */
/* *****/

typedef struct stop_tn62
{
    USHORT opcode; /* AP_STOP_TN62 */
    UCHAR reserv1[6]; /* Reserved */
    USHORT primary_rc; /* Primary RETURN_CODE */
    UCHAR reserv2[2]; /* Reserved */
    ULONG secondary_rc; /* Secondary RETURN_CODE */
    UCHAR reserv3[4]; /* Reserved */
    UCHAR key[8]; /* Key (ASCII) */
} STOP_TN62;

/**STRUCT-*****/

/**STRUCT+*****/
/* Structure: DEFINE_LOCAL_LU_TN62 */
/* */
/* Description: DEFINE_LOCAL_LU_TN62 verb control block */
/* *****/

typedef struct define_local_lu_tn62
{
    USHORT opcode; /* Verb operation code */
    /* AP_DEFINE_LOCAL_LU_TN62 */
    UCHAR reserv1[6]; /* Reserved */
    USHORT primary_rc; /* Primary RETURN_CODE */
    UCHAR reserv2[2]; /* Reserved */
    ULONG secondary_rc; /* Secondary RETURN_CODE */
    UCHAR reserv3[4]; /* Reserved */
    UCHAR key[8]; /* Key (ASCII) */
    UCHAR lu_name[8]; /* LU Name (EBCDIC type A) */
    UCHAR lu_alias[8]; /* LU Alias (ASCII) */
    UCHAR nau_address; /* Network Addressable Unit Address */
    /* 0 or 1-254 */
    UCHAR external_support; /* External support for sync point */
    /* AP_NONE x'00' */
    /* AP_SYNCPT_PROVIDER x'80' */
    /* AP_REMOTE_TP_PROVIDER x'40' */
    /* AP_SYNCPT_AND_REMOTE_TP x'CO' */
    UCHAR host_link_name[8]; /* Host link name */
    /* 0 or 1-8 bytes (EBCDIC type A) */
    UCHAR lu_model_name[7]; /* Self-defining dep LU model name */
    UCHAR reserv4[35]; /* Reserved */
    ULONG ip_address_mask;
    UCHAR register_to_dns;
} DEFINE_LOCAL_LU_TN62;

/**STRUCT-*****/

/**STRUCT+*****/
/* Structure: DEFINE_PARTNER_LU_TN62_OVERLAY */
/* */

```

```

/*                                                                 */
/* Description: DEFINE_PARTNER_LU_TN62_OVERLAY is used after all the */
/*             alt_alias_overlay structures (if any) in the         */
/*             DEFINE_PARTNER_LU VCB. The opcode field in the       */
/*             DEFINE_PARTNER_LU VCB must be set to DEFINE_PARTNER_LU_TN62. */
/*             ***** */
/*             ***** */

typedef struct define_partner_lu_tn62_overlay
{
    ULONG ip_address_mask;
    ULONG partner_ip_addr;
    UCHAR partner_hostname[TN62_MAX_HNAME];
} DEFINE_PARTNER_LU_TN62_OVERLAY;

/**STRUCT-***** */

/* ***** */
/* TN62 configuration API function prototypes */
/* ***** */
extern void APIENTRY TN62API(PVOID vcb);

/* ***** */
/* Define this header file to be C if this is included by a C++ source file */
/* ***** */
#ifdef __cplusplus
}
#endif

/* ***** */
/* End ifndef _TN62API_H_ */
/* ***** */
#endif

/* tn62api.h */

```

A.2 TCP62 Sample Program

```

#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include "tn62api.h"

static int quiet = 0;

/*
** ZERO: fill a field with hex zeros
*/
#define ZERO(var) memset(&(var), '\0', sizeof((var)))

/*
** BLANK: fill a field with ASCII blanks
*/
#define BLANK(var) memset(&(var), ' ', sizeof((var)))

/*
** STR2ARY: copy a null terminated string to an array, padding with
**          ASCII blanks, if necessary, and omitting the trailing NULL.

```

```

**          If the string is too long for the array, the excess is not
**          copied.
*/
#define STR2ARY(ary, str) \
{ \
    memset(ary, ' ', sizeof(ary)); \
    memcpy(ary, str, min(strlen(str), sizeof(ary))); \
}

/*
** ARY2STR: copy an array of characters to a string, stripping trailing
**          blanks and nulls and adding the null terminator. The string
**          is assumed to be large enough to hold the array.
*/
#define ARY2STR(str, ary) \
{ \
    int ARY2STRi; \
    for (ARY2STRi = sizeof(ary) - 1; \
        (ARY2STRi >= 0) && \
        ((ary)[ARY2STRi] == ' ' || (ary)[ARY2STRi] == '\0'); \
        ARY2STRi--) \
    { \
        ; /* null loop body */ \
    } \
    (str)[ARY2STRi + 1] = '\0'; \
    for (; ARY2STRi >= 0; ARY2STRi--) \
    { \
        (str)[ARY2STRi] = (ary)[ARY2STRi]; \
    } \
}

/*
** ASCII to EBCDIC translate table, from MYTEUSA
*/
static const unsigned char asc2BcdTable[] = {
    "\x00\x01\x02\x03\x37\x2D\x2E\x2F\x16\x05\x25\x0B\x0C\x0D\x0E\x0F"
    "\x10\x11\x12\x13\x3C\x3D\x32\x26\x18\x19\x3F\x27\x22\x1D\x35\x1F"
    "\x40\x5A\x7F\x7B\x5B\x6C\x50\x7D\x4D\x5D\x5C\x4E\x6B\x60\x4B\x61"
    "\xF0\xF1\xF2\xF3\xF4\xF5\xF6\xF7\xF8\xF9\x7A\x5E\x4C\x7E\x6E\x6F"
    "\x7C\xC1\xC2\xC3\xC4\xC5\xC6\xC7\xC8\xC9\xD1\xD2\xD3\xD4\xD5\xD6"
    "\xD7\xD8\xD9\xE2\xE3\xE4\xE5\xE6\xE7\xE8\xE9\xAD\xE0\xBD\x5F\x6D"
    "\x79\x81\x82\x83\x84\x85\x86\x87\x88\x89\x91\x92\x93\x94\x95\x96"
    "\x97\x98\x99\xA2\xA3\xA4\xA5\xA6\xA7\xA8\xA9\xC0\x4F\xD0\xA1\x07"
    "\x43\x20\x21\x1C\x23\xEB\x24\x9B\x71\x28\x38\x49\x90\xBA\xEC\xDF"
    "\x45\x29\x2A\x9D\x72\x2B\x8A\x9A\x67\x56\x64\x4A\x53\x68\x59\x46"
    "\xEA\xDA\x2C\xDE\x8B\x55\x41\xFE\x58\x51\x52\x48\x69\xDB\x8E\x8D"
    "\x73\x74\x75\xFA\x15\xB0\xB1\xB3\xB4\xB5\x6A\xB7\xB8\xB9\xCC\xBC"
    "\xAB\x3E\x3B\x0A\xBF\x8F\x3A\x14\xA0\x17\xCB\xCA\x1A\x1B\x9C\x04"
    "\x34\xEF\x1E\x06\x08\x09\x77\x70\xBE\xBB\xAC\x54\x63\x65\x66\x62"
    "\x30\x42\x47\x57\xEE\x33\xB6\xE1\xCD\xED\x36\x44\xCE\xCF\x31\xAA"
    "\xFC\x9E\xAE\x8C\xDD\xDC\x39\xFB\x80\xAF\xFD\x78\x76\xB2\x9F\xFF"
};

/*
** Asc2Bcd: convert characters from ASCII to EBCDIC
*/
static unsigned int
Asc2Bcd(unsigned char *d, /* Pointer to destination buffer */
        unsigned char *s, /* Pointer to source buffer */

```

```

        unsigned int n) /* Number of characters to convert */
{
    register unsigned int i;
    for (i = 0; i < n; i++) {
        *d++ = asc2BcdTable[*s++];
    }
    return n;
}

/*
** ASC2BCD: convert an array of chars from ASCII to EBCDIC, in place
*/
#define ASC2BCD(ary) (void) Asc2Bcd((ary), (ary), sizeof(ary))

/* EBCDIC to ASCII translate table, from MYTEUSA */
static const unsigned char bcd2AscTable[] = {
    "\x00\x01\x02\x03\xCF\x09\xD3\x7F\xD4\xD5\xC3\x0B\x0C\x0D\x0E\x0F"
    "\x10\x11\x12\x13\xC7\xB4\x08\xC9\x18\x19\xCC\xCD\x83\x1D\xD2\x1F"
    "\x81\x82\x1C\x84\x86\x0A\x17\x1B\x89\x91\x92\x95\xA2\x05\x06\x07"
    "\xE0\xEE\x16\xE5\xD0\x1E\xEA\x04\x8A\xF6\xC6\xC2\x14\x15\xC1\x1A"
    "\x20\xA6\xE1\x80\xEB\x90\x9F\xE2\xAB\x8B\x9B\x2E\x3C\x28\x2B\x7C"
    "\x26\xA9\xAA\x9C\xDB\xA5\x99\xE3\xA8\x9E\x21\x24\x2A\x29\x3B\x5E"
    "\x2D\x2F\xDF\xDC\x9A\xDD\xDE\x98\x9D\xAC\xBA\x2C\x25\x5F\x3E\x3F"
    "\xD7\x88\x94\xB0\xB1\xB2\xFC\xD6\xFB\x60\x3A\x23\x40\x27\x3D\x22"
    "\xF8\x61\x62\x63\x64\x65\x66\x67\x68\x69\x96\xA4\xF3\xAF\xAE\xC5"
    "\x8C\x6A\x6B\x6C\x6D\x6E\x6F\x70\x71\x72\x97\x87\xCE\x93\xF1\xFE"
    "\xC8\x7E\x73\x74\x75\x76\x77\x78\x79\x7A\xEF\xC0\xDA\x5B\xF2\xF9"
    "\xB5\xB6\xFD\xB7\xB8\xB9\xE6\xBB\xBC\xBD\x8D\xD9\xBF\x5D\xD8\xC4"
    "\x7B\x41\x42\x43\x44\x45\x46\x47\x48\x49\xCB\xCA\xBE\xE8\xEC\xED"
    "\x7D\x4A\x4B\x4C\x4D\x4E\x4F\x50\x51\x52\xA1\xAD\xF5\xF4\xA3\x8F"
    "\x5C\xE7\x53\x54\x55\x56\x57\x58\x59\x5A\xA0\x85\x8E\xE9\xE4\xD1"
    "\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\xB3\xF7\xF0\xFA\xA7\xFF"
};
/*
** Bcd2Asc: convert characters from EBCDIC to ASCII
*/
static unsigned int
Bcd2Asc(unsigned char *d, /* Pointer to destination buffer */
        unsigned char *s, /* Pointer to source buffer */
        unsigned int n) /* Number of characters to convert */
{
    unsigned int i;
    for (i = 0; i < n; i++) {
        *d++ = bcd2AscTable[*s++];
    }
    return n;
}

/*
** BCD2ASC: convert an array of chars from EBCDIC to ASCII, in place
*/
#define BCD2ASC(ary) (void) Bcd2Asc((ary), (ary), sizeof(ary))

/*
** UsageError: print usage instructions and exit
*/
static void UsageError(void)
{

```



```

printf("Usage:\n");
printf("\n");
printf("  TN62TEST [flags] operation\n");
printf("\n");
printf("    where flags is one or more of:\n");
printf("        -q  quiet mode - the only thing written to stdout\n");
printf("                is the blank-delimited primary and\n");
printf("                secondary return codes in hex.\n");
printf("        Useful for automated testing.\n");
printf("\n");
printf("    and operation is one of:\n");
printf("        START          fqcp_name\n");
printf("                        [cp_alias\n");
printf("                        [domain_name_suffix\n");
printf("                        [ip_address_mask\n");
printf("                        [connection_retry_secs\n");
printf("                        [unacked_dg_retry_secs\n");
printf("                        [unsent_dg_retry_secs\n");
printf("                        [inactivity_timer_secs\n");
printf("                        [conwait_secs]]]]\n");
printf("\n");
printf("        STOP\n");
printf("\n");
printf("        DEFINE_LOCAL_LU  lu_name\n");
printf("                        [ip_address_mask\n");
printf("                        [register_to_dns\n");
printf("                        [lu_alias]]]\n");
printf("\n");
printf("        DEFINE_PARTNER_LU partner_lu_name\n");
printf("                        [ip_address_mask\n");
printf("                        [partner_ip_address\n");
printf("                        [partner_hostname\n");
printf("                        [plu_alias\n");
printf("                        [plu_un_name\n");
printf("                        [max_mc_ll_send_size\n");
printf("                        [conv_security_ver\n");
printf("                        [parallel_sess_supp]]]\n");
printf("                        [alt_alias ...]]]]\n");
printf("        Notes:\n");
printf("        - the parameters are positional.  If the nth parameter is\n");
printf("        specified, then parameters 1 through n-1 must also be\n");
printf("        specified.\n");
printf("        - omitted trailing parameters take their default values.\n");
printf("        - the default value for a name is specified by \'.\'.\n");
printf("        - the default value for a number is specified by 0.\n");
printf("        - the default value for an IP address/mask specified by 0.0.0.0\n");
exit(0);
}

/*
** IpAddrStringToLong: convert an IP address string in dotted decimal
**                    notation to a long in big-endian format.  Error
**                    checking is very weak.
*/
static unsigned long IpAddrStringToLong(unsigned char *str)
{
    union
    {
        unsigned long asLong;

```

```

        unsigned char asByte[4];
    } result;
    unsigned int b[4];
    unsigned int i;
    sscanf(str, "%d.%d.%d.%d", &b[0], &b[1], &b[2],
    for (i = 0; i < 4; i++)
    {
        result.asByte[i] = (unsigned char) b[i];
    }
    return result.asLong;
}

/*
** Convert an IP address in long, big-endian format to a dotted decimal string
*/
static void IpAddrLongToString(char *str, unsigned long addr)
{
    union
    {
        unsigned long asLong;
        unsigned char asByte[4];
    } tempAddr;
    char tempStr[10];
    unsigned int i;
    str[0] = '\0';
    tempAddr.asLong = addr;
    for (i = 0; i < 4; i++)
    {
        sprintf(tempStr, "%u", tempAddr.asByte[i]);
        strcat(str, tempStr);
        if (i < 3)
        {
            strcat(str, ".");
        }
    }
    return;
}

/*
** Print the VCB return codes in human-readable format
*/
static void PrintVcbRcs(void *vcbP)
{
    char priStr[100];
    char secStr[100];
    switch (((APPC_HDR *) vcbP)->primary_rc)
    {
        case TN62_ERROR:
            strcpy(priStr, "TN62_ERROR");
            break;
        default:
            sprintf(priStr, "0x%4.4x", ((APPC_HDR *) vcbP)->primary_rc);
            break;
    }
    switch (((APPC_HDR *) vcbP)->secondary_rc)
    {
        case TN62_NODE_RUNNING_NO_ANYNET:
            strcpy(secStr, "TN62_NODE_RUNNING_NO_ANYNET");
            break;
    }

```

```

case TN62_CONFIGURATION_FILE_ERROR:
    strcpy(secStr, "TN62_CONFIGURATION_FILE_ERROR");
    break;
case TN62_NODE_NOT_STARTED:
    strcpy(secStr, "TN62_NODE_NOT_STARTED");
    break;
case TN62_NODE_START_INCOMPLETE:
    strcpy(secStr, "TN62_NODE_START_INCOMPLETE");
    break;
case TN62_PARAMETERS_NOT_USED:
    strcpy(secStr, "TN62_PARAMETERS_NOT_USED");
    break;
case TN62_NAME_GENERATION_ERROR:
    strcpy(secStr, "TN62_NAME_GENERATION_ERROR");
    break;
default:
    sprintf(secStr, "0x%8.8x", ((APPC_HDR *) vcbP)->secondary_rc);
    break;
}
printf("return code = (%s, %s)\n", priStr, secStr);
return;
}

/*
** Print the START_TN62 VCB in human-readable format
*/
static void PrintStartTn62Vcb(START_TN62 *vcbP)
{
    char tempStr[TN62_MAX_HNAME + 1];
    printf("opcode = AP_START_TN62\n");
    PrintVcbRcs(vcbP);
    memcpy(tempStr, vcbP->fqcp_name, sizeof(vcbP->fqcp_name));
    Bcd2Asc(tempStr, tempStr, sizeof(vcbP->fqcp_name));
    printf("fqcp_name = %17.17s\n", tempStr);
    printf("cp_alias = %8.8s\n", vcbP->cp_alias);
    IpAddrLongToString(tempStr, vcbP->ip_address_mask);
    printf("ip_address_mask = %s\n", tempStr);
    printf("vcbP->connection_retry_secs = %hu\n", vcbP->connection_retry_secs);
    printf("vcbP->unacked_dg_retry_secs = %hu\n", vcbP->unacked_dg_retry_secs);
    printf("vcbP->unsent_dg_retry_secs = %hu\n", vcbP->unsent_dg_retry_secs);
    printf("vcbP->inactivity_timer_secs = %hu\n", vcbP->inactivity_timer_secs);
    printf("vcbP->comwait_secs = %hu\n", vcbP->conwait_secs);
    ARY2STR(tempStr, vcbP->domain_name_suffix);
    printf("domain_name_suffix = %s\n", tempStr);
    return;
}

/*
** Print the STOP_TN62 VCB in human-readable format
*/
static void PrintStopTn62Vcb(STOP_TN62 *vcbP)
{
    printf("opcode = AP_STOP_TN62\n");
    PrintVcbRcs(vcbP);
    return;
}

/*
** Print the DEFINE_LOCAL_LU_TN62 VCB in human-readable format

```

```

*/
static void PrintDefineLocalLuTn62Vcb(DEFINE_LOCAL_LU_TN62 *vcbP)
{
    char tempStr[TN62_MAX_HNAME + 1];
    printf("opcode = AP_DEFINE_LOCAL_LU_TN62\n");
    PrintVcbRcs(vcbP);
    memcpy(tempStr, vcbP->lu_name, sizeof(vcbP->lu_name));
    Bcd2Asc(tempStr, tempStr, sizeof(vcbP->lu_name));
    printf("lu_name = %8.8s\n", tempStr);
    printf("lu_alias = %8.8s\n", vcbP->lu_alias);
    IpAddrLongToString(tempStr, vcbP->ip_address_mask);
    printf("ip_address_mask = %s\n", tempStr);
    switch (vcbP->register_to_dns)
    {
    case AP_NO:
        printf("register_to_dns = AP_NO\n");
        break;
    case AP_YES:
        printf("register_to_dns = AP_YES\n");
        break;
    default:
        printf("register_to_dns = 0x%2.2x\n", vcbP->register_to_dns);
        break;
    }
    return;
}

/*
** Print the DEFINE_PARTNER_LU_TN62 VCB in human-readable format
*/
static void PrintDefinePartnerLuTn62Vcb(DEFINE_PARTNER_LU *vcbP)
{
    ALT_ALIAS_OVERLAY *altAliasOverlayP;
    DEFINE_PARTNER_LU_TN62_OVERLAY *definePartnerLuTn62OverlayP;
    char tempStr[TN62_MAX_HNAME + 1];
    unsigned int i;
    printf("opcode = AP_DEFINE_PARTNER_LU_TN62\n");
    PrintVcbRcs(vcbP);
    memcpy(tempStr, vcbP->fqplu_name, sizeof(vcbP->fqplu_name));
    Bcd2Asc(tempStr, tempStr, sizeof(vcbP->fqplu_name));
    printf("fqplu_name = %17.17s\n", tempStr);
    printf("plu_alias = %8.8s\n", vcbP->plu_alias);
    memcpy(tempStr, vcbP->plu_un_name, sizeof(vcbP->plu_un_name));
    Bcd2Asc(tempStr, tempStr, sizeof(vcbP->plu_un_name));
    printf("plu_un_name = %8.8s\n", tempStr);
    printf("max_mc_ll_send_size = %hu\n", vcbP->max_mc_ll_send_size);
    switch (vcbP->conv_security_ver)
    {
    case AP_NO:
        printf("conv_security_ver = AP_NO\n");
        break;
    case AP_YES:
        printf("conv_security_ver = AP_YES\n");
        break;
    default:
        printf("conv_security_ver = 0x%2.2x\n", vcbP->conv_security_ver);
        break;
    }
    switch (vcbP->parallel_sess_supp)

```

```

{
case AP_NO:
    printf("parallel_sess_supp = AP_NO\n");
    break;
case AP_YES:
    printf("parallel_sess_supp = AP_YES\n");
    break;
default:
    printf("parallel_sess_supp = 0x%2.2x\n", vcbP->parallel_sess_supp);
    break;
}
altAliasOverlayP = (ALT_ALIASES_OVERLAY *) (((char *) vcbP) + sizeof(*vcbP));
for (i = vcbP->num_of_alt_aliases; i > 0; i--)
{
    printf("alt_alias = %8.8s\n", altAliasOverlayP->alt_plu_alias);
    altAliasOverlayP += 1;
}
definePartnerLuTn62OverlayP = (DEFINE_PARTNER_LU_TN62_OVERLAY *) ((char *)
                                                                    altAliasOverlayP);
IpAddrLongToString(tempStr, definePartnerLuTn62OverlayP->ip_address_mask);
printf("ip_address_mask = %s\n", tempStr);
IpAddrLongToString(tempStr, definePartnerLuTn62OverlayP->partner_ip_addr);
printf("partner_ip_addr = %s\n", tempStr);
ARY2STR(tempStr, definePartnerLuTn62OverlayP->partner_hostname);
printf("partner_hostname = %s\n", tempStr);
return;
}

/*
** Print the TN62 VCB in human-readable format
*/
static void PrintTn62Vcb(void *vcbP)
{
    switch (((APPC_HDR *) vcbP)->opcode)
    {
    case AP_START_TN62:
        PrintStartTn62Vcb((START_TN62 *) vcbP);
        break;
    case AP_STOP_TN62:
        PrintStopTn62Vcb((STOP_TN62 *) vcbP);
        break;
    case AP_DEFINE_LOCAL_LU_TN62:
        PrintDefineLocalLuTn62Vcb((DEFINE_LOCAL_LU_TN62 *) vcbP);
        break;
    case AP_DEFINE_PARTNER_LU_TN62:
        PrintDefinePartnerLuTn62Vcb((DEFINE_PARTNER_LU *) vcbP);
        break;
    default:
        printf("opcode = Unknown (0x%2.2x)\n", (((APPC_HDR *) vcbP)->opcode);
        break;
    }
    return;
}

/*
** CallTn62: call the TN62 config API
*/
static void CallTn62(void *vcbP)                /* Pointer to the vcb */
{

```

```

    if (!quiet)
    {
        printf("\nBefore calling TN62:\n");
        PrintTn62Vcb(vcbP);
    }
    TN62API((PV0ID) vcbP);
    if (!quiet)
    {
        printf("\nAfter calling TN62:\n");
        PrintTn62Vcb(vcbP);
    }
    else
    {
        printf("%4.4x %8.8x\n",
            ((APPC_HDR *) vcbP)->primary_rc,
            ((APPC_HDR *) vcbP)->secondary_rc);
    }
    return;
}

/*
** main
*/
int main(int argc, char *argv[])
{
    int argi;
    char operation[100];
    argi = 1;
    if (argc <= argi)
    {
        UsageError();
    }
    if (argv[argi][0] == '-')
    {
        if (argv[argi][1] == 'q')
        {
            quiet = 1;
        }
        else
        {
            UsageError();
        }
        argi++;
    }
    if (argc <= argi)
    {
        UsageError();
    }

    strcpy(operation, argv[argi]);
    argi++;
    strupr(operation);
    /*
    ** START_TN62 processing
    */
    if (!strcmp(operation, "START"))
    {
        START_TN62 vcb;
    }

```

```

ZERO(vcb);
vcb.opcode = AP_START_TN62;

if (argc <= argi)
{
    UsageError();
}

STR2ARY(vcb.fqcp_name, argv[argi]);
argi++;
ASC2BCD(vcb.fqcp_name);

BLANK(vcb.cp_alias);
if (argc > argi)
{
    if (argv[argi][0] != '.')
    {
        STR2ARY(vcb.cp_alias, argv[argi]);
    }
    argi++;
}

BLANK(vcb.domain_name_suffix);
if (argc > argi)
{
    if (argv[argi][0] != '.')
    {
        STR2ARY(vcb.domain_name_suffix, argv[argi]);
    }
    argi++;
}

vcb.ip_address_mask = 0;
if (argc > argi)
{
    vcb.ip_address_mask = IpAddrStringToLong(argv[argi]);
    argi++;
}

vcb.connection_retry_secs = 0;
if (argc > argi)
{
    vcb.connection_retry_secs = (USHORT) atoi(argv[argi]);
    argi++;
}

vcb.unacked_dg_retry_secs = 0;
if (argc > argi)
{
    vcb.unacked_dg_retry_secs = (USHORT) atoi(argv[argi]);
    argi++;
}

vcb.unsent_dg_retry_secs = 0;
if (argc > argi)
{
    vcb.unsent_dg_retry_secs = (USHORT) atoi(argv[argi]);
    argi++;
}

```

```

vcb.inactivity_timer_secs = 0;
if (argc > argi)
{
    vcb.inactivity_timer_secs = (USHORT) atoi(argv[argi]);
    argi++;
}

vcb.conwait_secs = 0;
if (argc > argi)
{
    vcb.conwait_secs = (USHORT) atoi(argv[argi]);
    argi++;
}

CallTn62(&vcb);
}

/*
** STOP_TN62 processing
*/
else if (!strcmp(operation, "STOP"))
{
    STOP_TN62 vcb;
    ZERO(vcb);
    vcb.opcode = AP_STOP_TN62;
    CallTn62(&vcb);
}

/*
** DEFINE_LOCAL_LU_TN62 processing
*/
else if (!strcmp(operation, "DEFINE_LOCAL_LU"))
{
    DEFINE_LOCAL_LU_TN62 vcb;

    ZERO(vcb);
    vcb.opcode = AP_DEFINE_LOCAL_LU_TN62;

    if (argc <= argi)
    {
        UsageError();
    }
    STR2ARY(vcb.lu_name, argv[argi]);
    argi++;
    ASC2BCD(vcb.lu_name);

    vcb.ip_address_mask = 0;
    if (argc > argi)
    {
        vcb.ip_address_mask = IpAddrStringToLong(argv[argi]);
        argi++;
    }

    vcb.register_to_dns = AP_NO;
    if (argc > argi)
    {
        if (!strcmp(argv[argi], "AP_YES"))
        {

```



```

        vcb.register_to_dns = AP_YES;
    }
    argi++;
}

BLANK(vcb.lu_alias);
if (argc > argi)
{
    if (argv[argi][0] != '.')
    {
        STR2ARY(vcb.lu_alias, argv[argi]);
    }
    argi++;
}
CallTn62(&vcb);
}

/*
** DEFINE_PARTNER_LU_TN62 processing
*/
else if (!strcmp(operation, "DEFINE_PARTNER_LU"))
{
#define PLU_ALIAS_ARGC 11
    unsigned int vcbSize;
    DEFINE_PARTNER_LU          *vcbP;
    ALT_ALIAS_OVERLAY          *altAliasOverlayP;
    DEFINE_PARTNER_LU_TN62_OVERLAY *definePartnerLuTn62OverlayP;

    vcbSize = sizeof(*vcbP);
    if (argc > PLU_ALIAS_ARGC)
    {
        vcbSize += sizeof(*altAliasOverlayP) * (argc - PLU_ALIAS_ARGC);
    }
    vcbSize += sizeof(*definePartnerLuTn62OverlayP);
    vcbP = malloc(vcbSize);
    if (argc > PLU_ALIAS_ARGC)
    {
        altAliasOverlayP = (ALT_ALIAS_OVERLAY *) (((UCHAR *) vcbP) + sizeof(*vcbP));
        definePartnerLuTn62OverlayP = (DEFINE_PARTNER_LU_TN62_OVERLAY *)
            (((UCHAR *) altAliasOverlayP) + sizeof(*altAliasOverlayP) *
            (argc - PLU_ALIAS_ARGC));
    }
    else
    {
        altAliasOverlayP = NULL;
        definePartnerLuTn62OverlayP = (DEFINE_PARTNER_LU_TN62_OVERLAY *)
            (((UCHAR *) vcbP) + sizeof(*vcbP));
    }

    memset(vcbP, '\0', sizeof(*vcbP));
    vcbP->opcode = AP_DEFINE_PARTNER_LU_TN62;

    if (argc <= argi)
    {
        UsageError();
    }
    STR2ARY(vcbP->fqplu_name, argv[argi]);
    ASC2BCD(vcbP->fqplu_name);
    argi++;
}

```

```

definePartnerLuTn620verlayP->ip_address_mask = 0;
if (argc > argi)
{
    definePartnerLuTn620verlayP->ip_address_mask = IpAddrStringToLong(argv&lbri");
    argi++;
}

definePartnerLuTn620verlayP->partner_ip_addr = 0;
if (argc > argi)
{
    definePartnerLuTn620verlayP->partner_ip_addr = IpAddrStringToLong(argv&lbri");
    argi++;
}

BLANK(definePartnerLuTn620verlayP->partner_hostname);
if (argc > argi)
{
    if (argv[argi][0] != '.')
    {
        STR2ARY(definePartnerLuTn620verlayP->partner_hostname, argv[argi&rb
    }
    argi++;
}

BLANK(vcbP->plu_alias);
if (argc > argi)
{
    if (argv[argi][0] != '.')
    {
        STR2ARY(vcbP->plu_alias, argv[argi]);
    }
    argi++;
}

ZERO(vcbP->plu_un_name);
if (argc > argi)
{
    if (argv[argi][0] != '.')
    {
        STR2ARY(vcbP->plu_un_name, argv[argi]);
        ASC2BCD(vcbP->plu_un_name);
    }
    argi++;
}

if (argc > argi)
{
    if (argv[argi][0] != '.')
    {
        sscanf(argv[argi], "%hu", &vcbP->max_mc_ll_send_size);
    }
    argi++;
}

vcbP->conv_security_ver = AP_N0;
if (argc > argi)
{
    if (!strcmp(argv[argi], "AP_YES"))

```

```

        {
            vcbP->conv_security_ver = AP_YES;
        }
        argi++;
    }

    vcbP->parallel_sess_supp = AP_NO;
    if (argc > argi)
    {
        if (!strcmp(argv[argi], "AP_YES"))
        {
            vcbP->parallel_sess_supp = AP_YES;
        }
        argi++;
    }

    while (argi < argc)
    {
        vcbP->num_of_alt_aliases = (UCHAR) (vcbP->num_of_alt_aliases + 1);
        STR2ARY(altAliasOverlayP->alt_plu_alias, argv[argi]);
        altAliasOverlayP += 1;
        argi++;
    }

    CallTn62(vcbP);
}
else
{
    UsageError();
}
return 0;
}

```

Appendix B. Special Notices

This publication is intended to help network specialists and administrators to install, configure and monitor nodes running eNetwork Communications Server for OS/2 Warp. The information in this publication is not intended as the specification of any programming interfaces that are provided by eNetwork Communications Server for OS/2 Warp. See the PUBLICATIONS section of the IBM Programming Announcement for eNetwork Communications Server for OS/2 Warp for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other

operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

Advanced Peer-to-Peer Networking	AIX
AnyNet	Application System/400
APPN	AS/400
BookManager	CICS
eNetwork	ES/3090
ESCON	Extended Services for OS/2
FFST/2	IBM
LAN Distance	NetView
Nways	OS/2
PROFS	RS/6000
S/390	SAA
System/370	System/390
SystemView	VTAM
WIN-OS/2	

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Appendix C. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

C.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see "How to Get ITSO Redbooks" on page 289.

- *IBM Communications Server for OS/2 Warp Version 4.0 Enhancements*, SG24-4587
- *IBM Communications Server for OS/2 Warp Version 4.1 Enhancements*, SG24-4916
- *IBM Communications Server for Windows NT Version 5.0*, SG24-2099
- *APPN Architecture and Product Implementation Tutorial*, SG24-3669
- *Personal Communications V4.x Interoperability and Problem Determination*, GG24-4457
- *IBM VTAM APPN Handbook*, SG24-4823
- *OS/2 Warp Server, Windows NT, and NetWare: A Network Operating System Study*, SG24-4786
- *IBM 2210 Nways Multiprotocol Router - IBM 2216 Nways Multiaccess Connector - Description and Configuration Scenarios Volume II*, SG24-4956

C.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
Lotus Redbooks Collection	SBOF-6899	SK2T-8039
Tivoli Redbooks Collection	SBOF-6898	SK2T-8044
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
RS/6000 Redbooks Collection (PDF Format)	SBOF-8700	SK2T-8043
Application Development Redbooks Collection	SBOF-7290	SK2T-8037

C.3 Other Publications

These publications are also relevant as further information sources:

- *eNetwork Communications Server for OS/2 Warp Quick Beginnings*, GC31-8189
- *eNetwork Communications Server for OS/2 Warp Network Administration and Subsystem Management Guide*, SC31-8181
- *eNetwork Communications Server for OS/2 Warp Response File Reference*, SC31-8187

- *eNetwork Communications Server for OS/2 Warp Command Reference*, SC31-8183
- *eNetwork Communications Server for OS/2 Warp Problem Determination Guide*, SC31-8186
- *eNetwork Communications Server for OS/2 Warp 32-Bit ACDI Programming Reference*, SC31-7182
- *eNetwork Communications Server for OS/2 Warp 32-Bit APPC Programming Guide and Reference*, SC31-8148
- *eNetwork Communications Server for OS/2 Warp 32-Bit Application Programming Guide*, SC31-8152
- *eNetwork Communications Server for OS/2 Warp 32-Bit Conventional LUA Programming Reference*, SC31-8149
- *eNetwork Communications Server for OS/2 Warp CPI Communications Reference Supplement*, SC31-8153
- *eNetwork Communications Server for OS/2 Warp Developer's Toolkit for Intelligent Adapters*, SC31-7183
- *eNetwork Communications Server for OS/2 Warp Developer's Toolkit for ANDIS*, SC31-7179
- *eNetwork Communications Server for OS/2 Warp 32-Bit System Management Programming Reference*, SC31-8151
- *eNetwork Communications Server for OS/2 Warp 32-Bit X.25 Programming Reference*, SC31-8150
- *eNetwork Communications Server for OS/2 Warp Guide to AnyNet SNA over TCP/IP*, GC31-8193
- *eNetwork Communications Server for OS/2 Warp Guide to AnyNet Sockets over SNA*, GC31-8191
- *eNetwork Communications Server for OS/2 Warp Guide to AnyNet LAN Gateway*, GC31-8320
- *eNetwork Communications Server for OS/2 Warp Frame Relay User's Guide and Reference*, GC31-8319

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at <http://www.redbooks.ibm.com/>.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

To get LIST3820s of redbooks, type one of the following commands:

```
TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
```

To get BookManager BOOKs of redbooks, type the following command:

```
TOOLCAT REDBOOKS
```

To get lists of redbooks, type one of the following commands:

```
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
```

To register for information on workshops, residencies, and redbooks, type the following command:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1998
```

For a list of product area specialists in the ITSO: type the following command:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
```

- **Redbooks Web Site on the World Wide Web**
<http://w3.itso.ibm.com/redbooks/>
- **IBM Direct Publications Catalog on the World Wide Web**
<http://www.elink.ibm.link.ibm.com/pbl/pbl>

IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibm.link.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (<http://www.redbooks.ibm.com/redpieces.html>). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** — send orders to:

In United States:	IBMMAIL usib6fpl at ibmmail	Internet usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications Publications Customer Support P.O. Box 29570 Raleigh, NC 27626-0570 USA	IBM Publications 144-4th Avenue, S.W. Calgary, Alberta T2P 3N5 Canada	IBM Direct Services Sortemosevej 21 DK-3450 Allerød Denmark
--	--	--

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States) or (+1)001-408-256-5422 (Outside USA)** — ask for:

Index # 4421 Abstracts of new redbooks
Index # 4422 IBM redbooks
Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com

- **On the World Wide Web**

Redbooks Web Site	http://www.redbooks.ibm.com/
IBM Direct Publications Catalog	http://www.elink.ibm.link.ibm.com/pbl/pbl

- **Internet Listserver**

With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibm.link.ibm.com with the keyword `subscribe` in the body of the note (leave the subject line blank).

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (<http://www.redbooks.ibm.com/redpieces.html>). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

First name	Last name
------------	-----------

Company

Address

City	Postal code	Country
------	-------------	---------

Telephone number	Telefax number	VAT number
------------------	----------------	------------

- Invoice to customer number

- Credit card number

Credit card expiration date	Card issued to	Signature
-----------------------------	----------------	-----------

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Index

A

- adaptive rate-based control
 - See ARB
- adaptive session-level pacing 104
- ANR 96, 105, 107
 - labels 96, 120, 125
 - network layer header 121
- ANR traces 264
- anrtstop 266
- anrtstrt 265
- APPN 90
 - See also APPN
 - control point 99
 - end node 90, 97
 - HPR boundary function 107, 110
 - network node 90, 97
 - VTAM 100
- APPNF 259
- APPNT 259
- ARB 104
 - flow/congestion control 104
 - receiver 113
 - sender 113
- authorized end nodes 98
- automatic network routing
 - See ANR

B

- bibliography 287
- BIND 95
- border node 101
 - extended border node 102
 - peripheral border node 102
- boundary function 95
- boundary node 99
- branch extender node 157, 160, 161, 163, 164, 165, 166, 169, 170
 - cascaded BrNN 166
 - configuration 166
 - connection networks 164
 - cryptography 165
 - design overview 161
 - DLUR 165
 - HPR 165
 - NDF file configuration 170
 - options 166
 - overview 157
 - requirements 160
 - response file 169
 - restrictions 163
- BrNN 157

C

- central resource registration 97
- client workstations 33
- cliff point 112
- CNN 93, 100, 101
- communication features 12
- composite network node
 - See CNN
- composite node 99, 100
- connection network 101, 125
 - HPR 123
- connectivity options and services 13
- control flows over RTP option 123
- COS 97, 107
- CP-CP session 100, 101, 106
 - HPR 129
- CPI-C for Java 19, 21, 22, 25, 26, 27
 - benefits 21
 - compiling Java programs 25
 - executing CPI-C for Java programs 26
 - feature install (FI) 21
 - implementation 19
 - installation 21
 - Java for OS/2 1.1.1 22
 - running JPing 27
 - troubleshooting 26

D

- default MLTG 127
- dial information 123
- DLC
 - HPR 120
- DLUR 165
- domain 96

E

- end node 97
 - authorized 98
 - unauthorized 98
- enhanced session address 96
- error recovery 125
- ESCON 109
- explicit MLTGs 127
- extended border node 99, 102

F

- FFST/2 258
- FID2 125, 128
 - PIU 109
- frame relay 128

G

getsense 58
utility 58

H

High-Performance Routing 87
 See also HPR
HOD 225, 227, 229, 231
 configuration 231
 overview 225
 request routing 229
 required software 227
Host On-Demand 225
host terminal emulations 11
HPR 87, 93, 96, 165
 addressing 96
 base 108
 capabilities control vector 126
 control flows over RTP 110
 HPR-only path 118
 link 126
 migration 93
 node 99
 options 103, 108
 overview 87
 SAP 128
 subnet 106, 107

I

ICN 99, 100
IEEE
 802.2 LLC Type 2 128
inactivity message 129
intermediate node 89
introduction 3
ISR 97, 105

K

knee point 112

L

LAN 128
LEN 88
 connection 88
 end node 88, 98
level 2 protocol identifier 128
level 3 protocol identifier 128
limited resource 125
 HPR 122
link
 HPR 126
 inactivity timer 129
 link-level error recovery 104, 105, 109, 128
 station 128

LLC commands 128
LLC responses 128
local address 95
local session identifier 95
logical unit
 See LU
low-entry networking
 See LEN
lowest-weight HPR-only route 118
LU 88

M

MLTG 104, 123
 ANR labels 125
 negotiation 127
 supported indicator 127
mobile partners 116
Multi-Link Transmission Groups
 See MLTG

N

NAU 94, 95
NCE 110
negotiation-proceeding XID3 126
net ID 94
 registry 94
network accessible unit
 See NAU
network address 95
network connection endpoint
 See NCE
network ID 94
network name 94
network node 97
network node server
 See NNS
network-qualified name 94
NHDR 120
NLP 109
NNS 97
node
 HPR node 102
Non-activation XID 125
non-disruptive path switch 103, 110, 115
number of retries 129

O

OS/2 client workstations 59
OS/2 workstations 59

P

parallel TGs 123
peripheral border node 99, 102
problem determination 257

programming support 11

R

rate

reply 114

request 114

re-sequence 104

release overview 3

resequencing 124

ring 0 traces 263

route

computation

HPR 107

setup messages 109

setup request 123

RSCV 107

RTP 96, 103, 110

connection 103, 107, 110

endpoints 125

functions for HPR 110

transport header 121

S

SAP 128

SATF 101

SDLC 109, 128, 141, 142, 144

device driver 144

installation 141

WAC adapter 142

installation 142

SDLC Support 141

send

timer 129

session

connector 95

identifier 95

initiation 95

single-link TGs 123

SNA 90

SNA Gateway 199, 200, 202, 203, 205, 207

configuration 200

DLUR PU 205

configuration 205

explicit workstations 203

host links 207

host LU pools 202

hosts 202

implicit workstations 203

support 199

SNA gateway support 10

SNA over TCP/IP 14

access nodes 14

gateway 14

SNA phone connect 13

Sockets over SNA 14

access nodes 14

gateway 14

SSCP 96, 99

SSCP-SSCP session 100

stationary partners 116

store-and-forward 124

subarea

nodes 99

subnet

HPR subnet 102

subnetwork 101

system services control point

See SSCP

systems network architecture

See SNA

T

TCP62 API 173, 174, 175, 187, 188, 192, 267

dynamic name generation 175

hosts files 188

monitoring 192

overview 173

sample configuration 187

sample program 267

writing TCP62 programs 174

TG 93

multilink 93

multiple 93

parallel 93

single-link TGs 93

TH 95

FID5 121

THDR 115, 120

TN3270E server 14, 213, 214, 217, 220

administration 220

configuration 214

in BrNN 213

parameters 217

topology database 107

topology subnetwork 102

trace 261, 265

services 261

spooling 265

traces 257, 263, 264

ANR 264

ring 0 263

transmission group

See TG

transmission priority 107

HPR 107

troubleshooting 257

information 257

TRS 125

U

UI frames 128

unauthorized end nodes 98

V

VR-TG 125
VRN 99, 101

W

Web-based administration 239, 240, 247
 FFST/2 247
 installation 240
 installing ICSS 4.2.1 240
 required hardware 240
 security 240
Windows 95 33
Windows NT 33
 access feature 33

X

X.25 109, 128
XID exchange
 negotiation proceeding 126
XID3 126
 capabilities control vector 126
 exchange 109

ITSO Redbook Evaluation

IBM eNetwork Communications Server for OS/2 Warp
SG24-2147-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at <http://www.redbooks.ibm.com>
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs?

Yes____ No____

If no, please explain:

What other redbooks would you like to see published?

Comments/Suggestions: **(THANK YOU FOR YOUR FEEDBACK!)**

